AD-A089 381 BATTELLE COLUMBUS LABS OH F/G 13/13
STRUCTURAL ANALYSIS VIA GENERALIZED INTERACTIVE GRAPHICS - STAG--ETC(U)
SEP 79 L E HULBERT, C SCOFIELD F33615-76-C-3125
UNCLASSIFIED AFFDL-TR-79-3074-VOL-2 NL

1 OF 2
AD
A089 381

AFFDL-TR-79-3074
Volume II

AD A089381

STRUCTURAL ANALYSIS VIA GENERALIZED INTERACTIVE GRAPHICS
STAGING
Volume II — User's Guide

L. E. HULBERT
N. D. GHADIALI
F. N. DROBOT
BATTELLE COLUMBUS LABORATORIES
505 KING AVENUE
COLUMBUS, OHIO 43201

SEPTEMBER 1979

TECHNICAL REPORT AFFDL-TR-79-3074, Volume II
Final Technical Report from June 1976 — September 1979

Approved for public release; distribution unlimited.

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Bernard H. Groomes
Project Engineer

Frederick A. Picchioni, Lt Col, USAF
Chief, Analysis & Optimization Branch
Structures & Dynamics Division

FOR THE COMMANDER

Ralph L. Kuster Jr., Col. USAF
Chief, Structures & Dynamics Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIBR*, W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFFDL-TR-79-3074 VOL[redacted]-2	2. GOVT ACCESSION NO.: AD-A089 381
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): STRUCTURAL ANALYSIS VIA GENERALIZED INTERACTIVE GRAPHICS - STAGING. VOLUME II. USER'S GUIDE. A087380	5. TYPE OF REPORT & PERIOD COVERED: TECHNICAL - FINAL report, JUN[redacted]76 - SEP[redacted]79
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): L. E. HULBERT, C. SCOFIELD	8. CONTRACT OR GRANT NUMBER(s): F33615-76-C-3125
9. PERFORMING ORGANIZATION NAME AND ADDRESS: BATTELLE, COLUMBUS LABORATORIES, 505 KING AVENUE, COLUMBUS, OHIO 43201	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 2401-02-10
11. CONTROLLING OFFICE NAME AND ADDRESS: AIR FORCE FLIGHT DYNAMICS LABORATORY (FBRA), WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433	12. REPORT DATE: SEP[redacted]79
	13. NUMBER OF PAGES: 174
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

INTERACTIVE GRAPHICS
FINITE ELEMENT MODELS
STRUCTURAL ANALYSIS
MESH GENERATION
COMPUTER AIDED DESIGN
COMPUTER AIDED ANALYSIS

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

STAGING (STructural Analysis via Generalized INteractive Graphics) has been developed to give engineers an interactive graphics system for constructing and studying finite element models and for reviewing the results of a finite element analysis.

This volume gives details of the installation and maintenance procedures for STAGING on CDC, CYBER, and 6000 computer systems. The structure of STAGING is detailed and purpose of each subroutine is described.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWARD

This final report was prepared by the Columbus Laboratories of Battelle Memorial Institute, Columbus, Ohio, for the Structures and Dynamics Division, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio. The work was performed under Contract No. F-33615-75-C-3125, which was initiated under Project No. 2401, "Structures and Dynamics", Task No. 02, "Design and Analysis Methods for Aerospace Vehicle Structures". Initially, Mr. L. Bernier (FBR) was the AFFDL project engineer for this effort, after which Mr. B.H. Groomes (FBR) was assigned the responsibility.

STAGING, as described in this report, represents a three-year combined Air Force-Navy effort, with specific support and contributions from Dr. Charles P. Poirier, Chief, Scientific Systems Analysis Branch, ASD Computer Center, Wright-Patterson Air Force Base, Ohio, Messers. James M. McKee and Michael E. Golden, Computation Mathematics and Logistics Department, Code 1844, Mr. Paul Mayer and Miss Jane A. Figula, Structures Department, Code 1730.5, The David W. Taylor Naval Ship Research and Development Center, Bethesda, Maryland. The technical graphics expertise of these government researchers are gratefully acknowledged.

The report consists of four volumes. Volume I, "Final Summary Report", presents an overview of the capabilities of the STAGING (STructural Analysis via Generalized INteractive Graphics) system. Volume II, "Users Guide", gives detailed instructions on how to use STAGING for finite element analysis. Volume III, "System Manual", describes the internal structure of STAGING and details procedures for installation and Maintainance of the System on CDC CYBER and 6000 series mainframe computers. Volume IV, "Appendices to the System Manual", includes lists of STAGING procedures, loader directives and cross-referenced tables of all entry names that occur in STAGING.

The program manager of this development was Dr. L. E. Hulbert of the Transportation and Structures Department. He was supported by N. D. Ghadiali of the same department and by a number of specialists from the Computer, Information Systems, and Education Department including:

E. Edwards | K. Cadmus
D. Kasik | C. Scofield
W. Young | F. Drobot

Bill Young contributed significantly to the preparation of this volume.

The work reported herein was conducted during the period of June 28, 1976 through June 1979. Some work on STAGING was carried out under contract F33615.

The present report was submitted for publication in June, 1979.

SUMMARY

This manual describes the use of STAGING, a computer based system for supporting structured analysis through interactive graphics. In particular, it introduces the user to how STAGING operates and explains the array of functions, subfunctions, and options available in STAGING.

Chapter 1 presents most of the underlying interaction concepts necessary for using STAGING. It describes user prerequisites, the operating environment, user interfaces, the data base organization, getting started, and recovering from error. The topics therein are useful throughout the rest of the manual, and the beginning user is encouraged to be familiar with them.

Chapters 2 through 7 describe the logical use of STAGING in solving analysis problems. These steps include generating a model data base, displaying a finite element model, modifying a picture, editing model data, interfacing to analysis programs, and postprocessing analysis results.

Chapter 8 contains several general utilities which are useful throughout STAGING. Some characteristics of this manual are noteworthy. The organization of this manual matches the process a user employs to solve real problems using STAGING. Furthermore, this manual documents the capabilities in the same organization and structure as they occur within STAGING. Thus, it is a roadmap to the hundreds of functions, subfunctions, and options in STAGING.

The table of contents of this manual reflects the organization of both the manual and the system. All functions and subfunctions are identified in section titles with all capital letters. This can serve as a quick reference since the structure matches the system.

Chapters 2 through 8 each contains a summary as the last section. These summaries briefly describe the capabilities discussed in the chapter and present diagrams of the functions, subfunctions, and options. Thus, they provide the user with an instant roadmap throughout the STAGING System.

Chapter 6 contains sections which describe how to execute specific analysis conversion programs (see Section 6.3). Descriptions of individual conversion routines are similar in format and are modular within this manual; each sections begins on a new page. Therefore, user developed conversion programs can be documented and easily inserted within this manual.

Once the user is familiar with the concepts and the operation of STAGING, this manual will only be needed occasionally. The user can create a useful, short reference guide by extracting and duplicating selected sections from this manual. There are several possible extractions, and each should be dictated by user requirements. Below is one suggested extraction for the general experienced user.

To Create the STAGING Instant User Guide	Extract these from the STAGING User Manual
Contents	Contents
Chapter 1	All Sections
Chapter 2	Section 2.5
Chapter 3	Sections 3.1, 3.8
Chapter 4	Section 4.14
Chapter 5	Sections 5.1, 5.5
Chapter 6	Section 6.3
Chapter 7	Section 7.5
Chapter 8	Section 8.8

The first page of each chapter and all diagrams and tables within the selected sections should also be extracted.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

CHAPTER 2
GENERATING STAGING DATA BASES

Page

CHAPTER 3
DISPLAYING FINITE ELEMENT MODELS

TABLE OF CONTENTS (Continued)

CHAPTER 4
MODIFYING THE PICTURE

CHAPTER 5
EDITING MODEL DATA

TABLE OF CONTENTS (Continued)

CHAPTER 6
INTERFACING TO ANALYSIS PROGRAMS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

Page

CHAPTER 1

INTRODUCTION

This section of the STAGING User Manual describes the environment in which the system operates and how the user interacts with the system. It presents information and concepts which are useful throughout succeeding chapters. The STAGING user is encouraged to read and understand this section of the manual before proceeding with other sections and before attempting to use the system. It is to be noted that the word system is to imply "specific STAGING computer code" and will appear throughout the discussions.

1.1 User Prerequisites

The STAGING User Manual assumes that the reader has prior knowledge of his computer system and its operating environment. The reader must be familiar with CDC 6000 series computers, the NOS/BE operating system, and the INTERCOM timesharing system. Information in these areas can be found in CDC documentation or installation published literature.

Some knowledge of computer graphics is also assumed, but decriptive text is included when graphics knowledge is integral to understanding and using STAGING. The reader should be familiar with TEKTRONIX computer graphics terminals. Some information describing these terminals is included in the next section.

STAGING is primarily intended to be applied to finite element analysis applications. Therefore, the most comprehensive and extensive use of STAGING requires prior experience or training in structural or dynamic analysis procedures and terminology. The general graphics capabilities can be applied to any user problem requiring such capabilities. There are specific capabilities which are only applicable to finite element analysis. The STAGING User Manual only describes the use of the system when applied to finite element models, and does not include specifics about the engineering application. Some details are provided for application specific software (such as finite element analysis programs.)

1.2 Operating Environment

STAGING is an interactive computer graphics interface implemented on the CDC 6000, CYBER 70 and CYBER 170 series computers and operated on the TEKTRONIX 401X series computer graphics terminals. It functions under the NOS/BE 1.3 operating system (current operating system level 481, FTN extended version 4.7 level 476 and Loader level 476) and is a normal INTERCOM interactive job. STAGING can be operated to communicate in the range of 30 characters per second (300 baud) to 960 characters per second (9600 baud).

1.2.1 Graphics Hardware

The TEKTRONIX 401X series computer graphics terminals include the 4010, 4012, 4014, and 4016. In addition to these graphics terminals, STAGING can be used with a TEKTRONIX 4006. Other computer graphics terminals which emulate, or act like, a TEKTRONIX 4014 can be used. There are several commercially available graphics terminals with this capability.

The TEKTRONIX 401X series terminals are direct view storage terminals (DVST). This technology impacts use of STAGING. The most notable consequence to the user is the loss of time caused by screen erasures. Since DVST's cannot selectively erase specific areas on the screen, the entire screen must be erased as the picture changes. When the screen is erased, a bright flash occurs.

Table 1.1 lists some of the physical characteristics of the TEKTRONIX 401X graphics terminals. The TEKTRONIX 4014 with Enhanced Graphics Module (EGM) and the 4016 provide an additional capability of dot-dash lines. These are used primarily in deformation or displacement plots and X-Y plots.

The TEKTRONIX 401X series terminals allow the user to select STAGING menu entries or parts of his model through the use of thumb wheel controlled crosshairs. The crosshairs appear on the screen as vertical and horizontal lines extending across the screen. Each of two thumb wheels independently controls the movement of the vertical or horizontal line. The intersection of these lines is the point or location of the user selection. The user signals his selection by positioning the crosshairs and typing any keyboard character except the return key. (Some TEKTRONIX terminals require a return to be typed after the character has been typed; this is a terminal option which can be changed to eliminate the requirement.) More detail on the use of the crosshairs for graphics input in STAGING is contained in Section 1.3.

The TEKTRONIX 4006 does not provide graphics input. Although STAGING can operate on this terminal, some of the interactive capabilities of STAGING are sacrificed.

TABLE 1.1. CHARACTERISTICS OF TEKTRONIX TERMINALS

	Hard Copy Unit Compatibility	Keyboard and Character Set	Alpha Modes			Graphics Mode			Graphics Mode with Enhanced Graphics Module			Graphic Input Mode (GIN)
			Char. Matrix	Char/ Line	Lines/ Screen	Addressable Points	Displayable Points	Vector Type	Addressable Points	Displayable Points	Vector Types	Addressable Points
4006-1	4631											Not Applicable
		64 CH.ASCII	5 x 7	75	35	1024 x 1024	1024 x 780	Solid Line	Not Applicable			3-1023 x 0-780
4012	4631	96 CH.ASCII Auto Repeat 96 CH.ASCII 88 CH.APL Auto Repeat	7 x 9	74	35	1024 x 1024	1024 x 780	Solid Line	Not Applicable			3-1023 x 0-780
4014-1	4631	96 CH.ASCII Auto Repeat 88 CH.APL	7 x 9	74 81 121 133	35 38 58 64	1024 x 1024	1024 x 780	Solid Line	4096 x 4096	4096 x 3123	Solid Dotted Short Dash Long Dash Dot-Dash Also Point Plot Incremental Plot	3-1023 x 0-780

1.3 User Interface

STAGING is indeed an interactive graphics interface between the user and his finite element analysis program and uses menus of selectable options as the primary method of interaction. Possible options are presented to the user as a list of words, or **buttons**. The user selects a desired button and the system responds with some visual change to the picture and presents the next, or possibly the same, list of buttons.

The following sections describe the screen organization, menu format and the type of user interactions which could occur during a STAGING session. Most of this information is generic to subsequent chapters.

1.3.1 Screen Organization

The screen is divided into three areas as illustrated in Figure 1.1. The DISPLAY AREA is used for displaying pictures and textual output. The MENU AREA is used for presenting menus and prompt messages. This area is also used to present templates for input data which are to be typed by the user. The DATA INPUT AREA is used by the user to enter input data requested by the system.

1.3.2 Selecting Menu Buttons

Every menu consists of two parts: a prompting message and a list of one or more buttons. (See Figure 1.2 for an example of a menu.) The prompting message provides guidance in what the list of buttons are used for or what the system expects the user to do next. Associated with each menu button is a letter.

Each button in the menu can be selected in one of two ways. The crosshairs may be positioned (via the thumb wheels) so that they intersect over the button; depressing the space bar on the keyboard indicates the user selection. After the space bar is depressed the crosshairs will disappear. The system acknowledges the selection by rewriting the selected button, causing the button to be temporarily highlighted. If the crosshairs are not correctly positioned over the button, they will

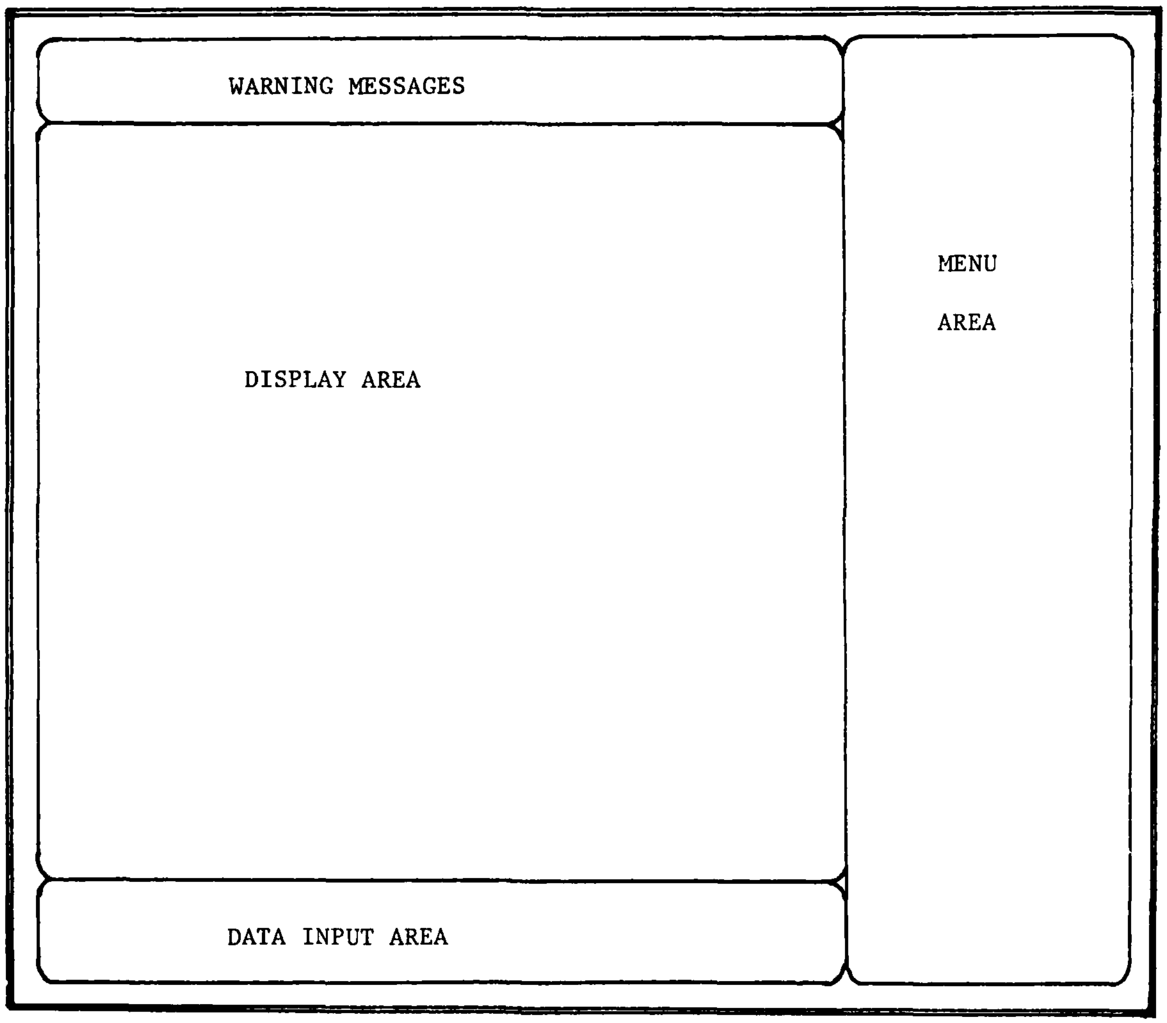


FIGURE 1.1. SCREEN ORGANIZATION

A. GLOBAL
B. REDRAW
C. ERASE

CHOOSE PORTIONS OF
MODEL TO BE DISPLAYED

D. STRUCTURES
E. SUBSTRUCTURES
F. ELEMENTS
G. NODES
H. RESTORE LAST DATA BA
I. SAVE CHANGED DATA BA
J. ERASE SCREEN
K. RETURN

FIGURE 1.2. SAMPLE MENU

reappear after a slight pause and the user must reposition the crosshairs more precisely.

Alternatively, each button can be selected by depressing the keyboard character for the letter associated with the desired button. When selecting buttons via the keyboard, the user should position the crosshairs out of the MENU AREA. The system first determines whether the crosshair intersection is a valid selection. If it is, the keyboard character is ignored, if not, the character is analyzed. If the character key is **invalid**, the crosshairs reappear after a slight pause. Note, however, that when a button is selected from the keyboard, the system does not rewrite the selected button to acknowledge the selection.

In summary, the system allows menu selection via both the crosshairs and the keyboard. The crosshairs take precedence over the keyboard when they intersect in the MENU AREA. As a level of redundancy, the novice user may wish to position the crosshairs **and** depress the keyboard character for the button letter.

Each menu selection determines the next menu to be displayed. *The user can imagine that menus are organized as a tree.* The next menu may be the next branch in the tree, the same menu, or a previous menu. At any time only **one** menu is active; that is, the user can only select from **one** set of menu buttons, even though more than one menu is displayed in the MENU AREA.

If a user returns to a previously displayed menu still on the screen (but not active), the system redisplays the menu in the same position. If not, new menus are displayed in the available space in the MENU AREA. When there is no space available, the system erases the screen and begins at the top. If there is a picture on the screen, the system requests the user to select one of two buttons before it erases the screen; REDRAW or ERASE. These buttons are located at the top of the MENU AREA and are always present. REDRAW erases the screen, redraws the picture in the DISPLAY AREA, and displays the new menu in the MENU AREA. ERASE erases the screen and only displays the new menu. When ERASE is selected, the user has not lost the picture. If REDRAW is selected after ERASE, the picture will be redrawn. Thus REDRAW and ERASE are always active, and the user may select them at any time. The GLOBAL button is

also located above the menu area and is discussed in Chapters 4 and 8.

On occasions, menus are presented with some of the buttons missing (See Figure 1.3) The system hides a button when selection of that button is invalid for some reason. In this example, the ERASE SCREEN button is not presented if there is no picture on the screen. The SAVE CHANGED DATA BASE and RESTORE DATA BASE are not shown until the initial data base has been changed (Compare with Figure 1.2) As another example, a DISPLAY ELEMENTS button would not be presented if all elements in the data base are presently being displayed. This feature helps the user guard against selecting invalid menu buttons.

As noted above, when the user returns to a displayed menu, the system redisplays the menu in the same position within the MENU AREA. The reasons for hiding a button are dynamically checked and, therefore, may change between presentations of a given menu. In one case, previously hidden buttons become available and will be displayed where previously there was not a button. The confusing case occurs when a previously presented button becomes invalid and is hidden. Because of the DVST technology (see Section 1.1.2) the button name still appears on the screen even though it cannot be selected by the user. If the user becomes confused as to which buttons are active in a list that has been repainted he is advised to select REDRAW (or ERASE) in order to have ONLY the active menu and its active buttons displayed.

Selection of an active button can cause one of a number of actions.

- o a new menu level is presented (picking a RETURN button returns the user to the next higher menu level)
- o STAGING will take the indicated action
- o STAGING may present a list of items from which the user is to make a selection (see Section 1.3.3)
- o STAGING will request that the user identify some part(s) of the displayed picture (See Section 1.3.4)
- o STAGING may request the user to enter some input data (see Section 1.3.5)

A. GLOBAL
B. REDRAW
C. ERASE

CHOOSE PORTIONS OF
MODEL TO BE DISPLAYED

D. STRUCTURES
E. SUBSTRUCTURES
F. ELEMENTS
G. NODES

K. RETURN

FIGURE 1.3. MENU WITH HIDDEN BUTTONS

1.3.3 Selecting From Listed Items

At certain points, STAGING will present a list of items from which the user is asked to make a selection. For example, if the user wishes to edit an element, he is presented a list of element attributes and directed to select the element attribute he wishes to change. These lists are displayed in the MENU AREA on the terminal screen. Such lists have no letter designation and must be picked by using the crosshairs. They cannot be picked from the keyboard and thus cannot be selected when a TEKTRONIX 4006 is being used.

With every such list a button is provided that completes the picking action and initiates the processing of the picked items. This button is usually called "NEXT" or "RETURN" and is designated in the prompting message written above the list of items. The prompting message will have the form "PICK ATTRIBUTES THEN NEXT".

Utilization of a separate button that completes the picking action protects the user from having STAGING process incorrectly selected items.

When a selection is made of a list item, STAGING acknowledges the selection by rewriting the selected item, causing it to be temporarily highlighted in the same way that a menu button is highlighted. All picks of this type act as on-off toggle switches. If the user mistakenly picks the wrong item he may deactivate his selection by picking the item a second time before picking the indicated menu button that completes the action (this button is usually "NEXT" or "RETURN").

In some cases STAGING requires that only one item may be picked. For such cases, if the user initially picks the wrong item, picking the correct item on a second try will deactivate the first pick, since STAGING will recognize only the last pick that the user makes before he completes the action.

In those cases for which STAGING allows more than one item to be selected at a time all selected items are qued up and will be processed when the "NEXT" or "RETURN" button is picked. In these cases the user must deactivate each incorrect pick by picking the item a second time. As noted, this picking process is equivalent to an on/off toggle switch.

Every item picked an odd number of times is active. Every item picked an even number of times is inactive. This approach helps the user to easily correct erroneous picks before STAGING processes the picked items.

1.3.4 Identifying Picture Parts

Several functions require the user to identify parts of the picture, or items (such as elements or nodes), to be processed or manipulated. One means of accomplishing this identification is by using the crosshairs.

An item is identified by locating the intersection of the crosshairs over the item and depressing the space bar on the keyboard. For line drawings, the crosshairs must be on the line; for characters, the crosshairs must be within the space of the character. The system acknowledges a valid selection by retracing the item, causing it to be temporarily highlighted. Although the system exercises some tolerance in interpreting which item the user is selecting, the crosshairs must be close to the desired item.

If the user misses picking the desired item, or any item, STAGING will return the crosshairs after a short pause to allow the user to more precisely locate the crosshairs. If the user picks the wrong item (as indicated by the highlighted response) he may deactivate this pick by picking the item a second time. Thus picking items from the displayed picture can be treated as an on/off toggle switch in the same way as picking from menu lists (see Section 1.3.3).

When STAGING requires only one picture item, it will process only the last item picked and the user needs to make sure that the last item is the one he wants. In other cases, where STAGING allows more than one picture item to be picked at a time, the user must make sure that incorrect picks must be "toggled" to "OFF". In certain cases, after an incorrect pick is turned off, STAGING will not recognize any subsequent picks. In these cases, the user must complete the picking action to process those items picked before the incorrect pick was made and then reinitiate the picture selection to select the remaining desired items. As in the case of picking menu items, the user indicates the end of his

selection process by selecting the displayed button (usually NEXT or RETURN) which is identified in the prompting message associated with the picture selection request. Selecting this button without making a picture item selection constitutes a simple no-action RETURN to the next higher menu level.

There are special considerations whenever the user is identifying picture parts. It may not be possible to unambiguously identify an item. This is true when lines or characters are concident. In three dimensional models, the system interprets the selection as the "most forward item". For two dimensional models, the system interprets the selection as the "first drawn item". The ambiguity of coincident lines can be reduced or eliminated by rotating the model (see Section 4.11) or shrinking elements (see Section 4.6).

Alternative methods may be used for identifying picture parts. These include typing in the part name or number. Also parts may be identified by selecting all parts with an attribute value specified by type in. The applicability of these methods depends on the type of item. (These alternatives must be used when the graphics terminal, such as the TEKTRONIX 4006, does not have crosshairs.)

It should be noted that if the user erases the screen, he cannot graphically locate any picture items. Although the system will interpret selections, the items are not visible until after they have been selected. The user can select REDRAW to have the picture drawn, then proceed with identifying parts. Alternatively, with a picture on the screen, if the system does not acknowledge an item pick by redrawing it, the picture on the screen may contain deleted items. The user can use REDRAW to make the picture current.

1.3.5 Entering Input Data

A number of STAGING functions require data to be entered from the keyboard. When the system requests input data, it displays a data input template in the MENU AREA and draws a data input box in the DATA INPUT AREA. As the user types data from the keyboard, it appears within the data input box.

The data input template describes the data items the user may enter. Each data item has three parts: a letter, a keyword, and, if available, the last stored value for the data item. The user may enter data either positionally, by letter, by keyword, or any combination of these. In any case, each data value is separated by commas. Data values are identified positionally by entering them in the same order as the data input template. Blank or null data values are indicated by successive commas. When data values are entered by either letter or keyword, an equal sign (=) is used. For example, 'A=data value' or 'KEYWORD=data value' are sample input data specifications.

Numeric input data can be entered as integers, floating point numbers, or exponentially formatted numbers. In floating and exponential numbers, a leading digit **must** be entered before the decimal point. That is, '.752' or '.32E-5' are invalid; '0.752' and '3.2E-6' are valid. Certain functions request a range of numbers. A range of numbers can be entered (in any format) as two (2) numbers separated by the string 'to' or 't'. For example, '10to20' and '10t20' are valid and equivalent. All numbers and number ranges are limited to ten (10) digits and special characters (including E, +, -, and decimal points). Valid examples include 5000T10000 or 5.E03T1.E4.

Alphanumeric input data are entered as strings of characters or digits; no special marks (such as quotes) are required. For certain data input templates, data values may be specified with reserved words, such as MINIMUM, MAXIMUM, AUTO, CENTER, and INFINITY. In general, these reserved words also have abbreviations, such as MIN, MAX, C and I. Reserved words are applicable **only** when the specific input data request allows them. Thus the user must refer to specific functions when determining valid input data specifications.

STAGING displays data input templates in groups of ten or less items. If the input data request includes more than ten data items, subsequent groups of ten data items are displayed after the user enters a carriage return signaling the end of data input in response to the first group.

If no data values are to be entered, the user may simply enter a carriage return. (Certain functions require a space to be entered, followed by a carriage return).

CAUTION: If the user incorrectly enters data from the keyboard, STAGING will give an error message and require a correction be made on this input before the user is allowed to proceed. If the user cannot amend his input to a form allowed by STAGING, his only recourse may be to enter the error recovery package (with %A) to escape from his predicament.

1.4 STAGING Data Base

The STAGING Data Base provides the medium for storing information describing a finite element model. All modules in STAGING use the data base for storage and communication. Although the user does not need to know in detail how the data base operates, he should have some familiarity with its general structure to help understand how to use STAGING to store, retrieve, and edit his data.

1.4.1 Data Base Organization

The data base is divided into five levels: structures, substructures, elements, nodes, and tables. The first four levels are illustrated in Figure 1.4. The fifth level is for storing alphanumeric data not specifically associated with the structure's elements and nodes.

There is no limitation on the number of items that may be defined on each of the five levels. Referring to Figure 1.4, note the entire model is considered to be a structure, while the wing is considered to be a substructure. Additional substructures are present in this model (tail, fuselage, node, and pods) but are not illustrated. The data structure is hierarchical in nature; that is, substructures are made up of elements, and elements are defined by nodes. Elements and nodes have specifically defined contents: nodes describe points in space, and elements describe their connectivity.

A substructure, on the other hand, may be user-defined to contain any set of elements in the data base. For example, one might want to have a substructure that contains all the elements, another that contains only elements made of one material type, and yet another that contains only elements in the area with lower-left corner (0,0) and upper- right corner (2,1). Creation of a substructure is described in Section 5.3.1.

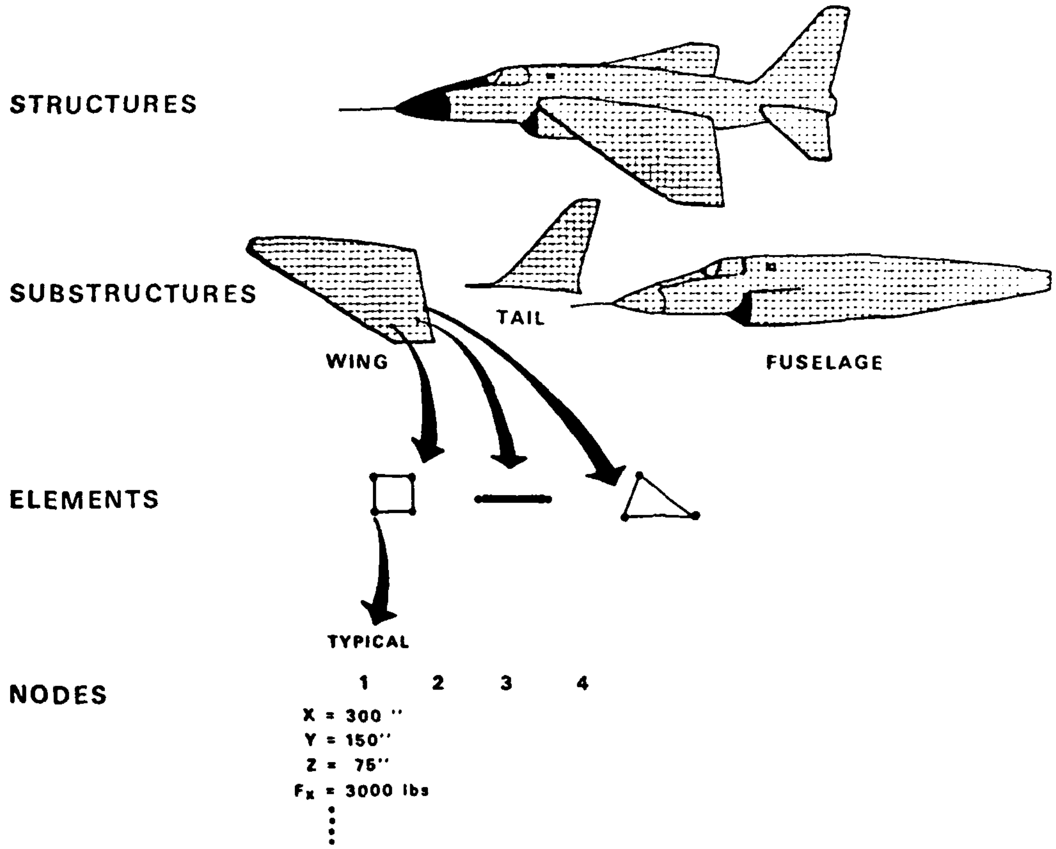


FIGURE 1.4. STAGING DATA BASE LEVELS

The first two levels in the data base (structures and substructures) permit names that are 40 characters long. This allows the structure or substructure to be named with a meaningful phrase.

The next two levels of the data base (elements and nodes) use conventional finite element numbers. The numbers must be positive integers.

Each of the nodes and elements defined in a data base may have associated data stored in arrays called attribute lists. The type of attribute assigned to entries at each level may be different.

The fifth level of the data base (tables) contains arrays which have unique table names, describing their format and dimensionality. Geometry independent information is stored in the tables level of data base as a separate list.

One of the basic objectives of STAGING is to make it as simple as possible to interface to a variety of analysis programs. The mechanism for accomplishing this is through data base conversion routines. These routines are set up to automatically transform data from an external source into a STAGING data base format. These data may be generated by preprocessors or may be contained on card decks.

After STAGING is used to view and correct the geometry of the model, a data base conversion routine is used to extract the appropriate information from the data base and transform it into an input file for the user's finite element program. Once the user has conducted the analysis of the problem, a third data conversion routine is used to transform the results of the analysis into the data base for interpretation and evaluation. The use of existing data conversion routines is discussed in Section 6.3. Directions for developing new data conversion routines is discussed in Sections 2.4, 6.1, and 6.2. Figure 1.5 illustrates the way that information can be transferred into and out of the STAGING data base. Because digitizer software is site dependent, no digitizer package is currently interfaced to STAGING. The user can interface his digitizer to STAGING by writing an appropriate data conversion routine as described in Section 2.4. Similarly, the user may have an automated mesh generation system in use. The output of such a generator may be inserted in the STAGING data base through a data conversion routine. At present, the DRAFTING node and element generator directly inserts data in the STAGING

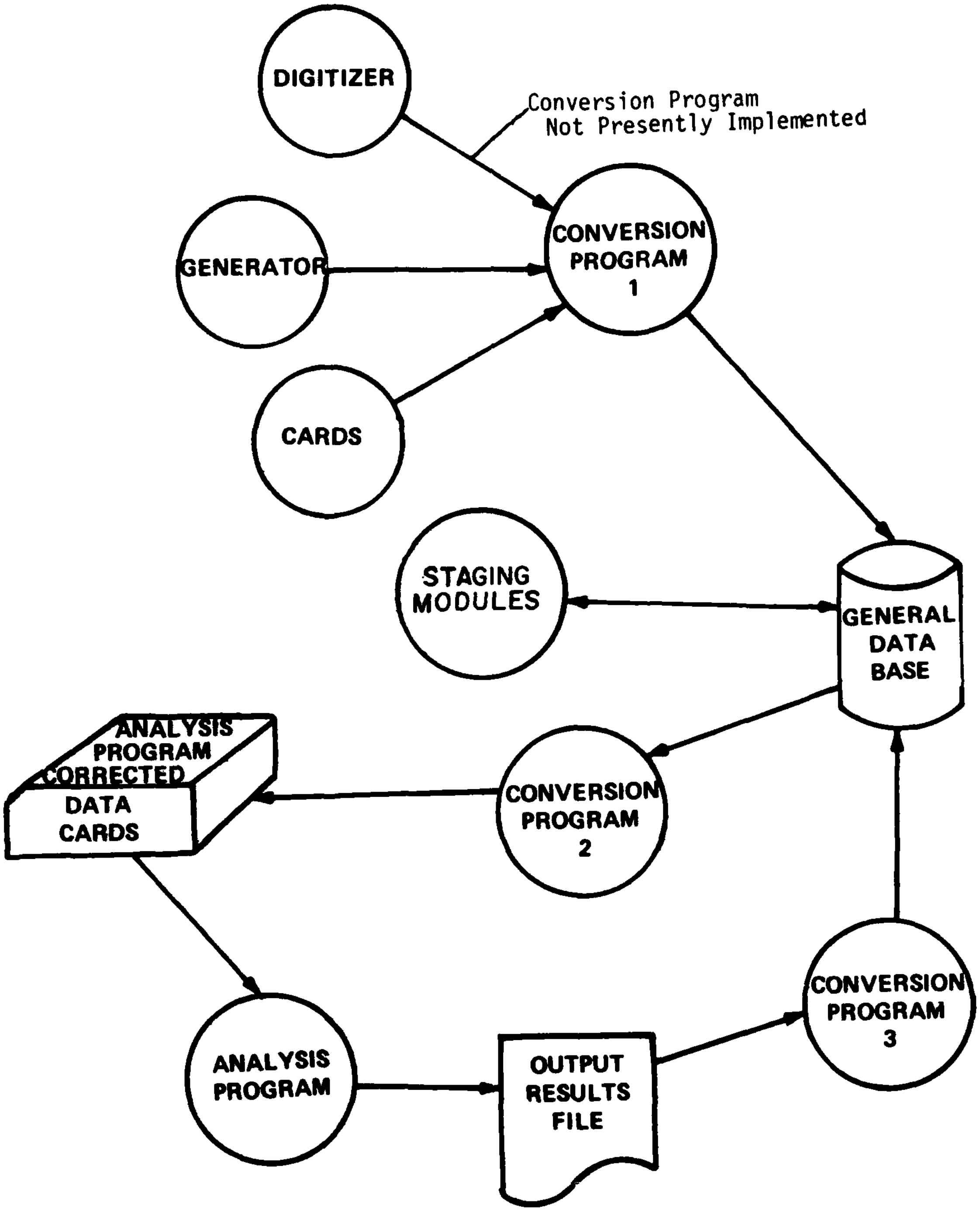


FIGURE 1.5. INFORMATION FLOW IN STAGING

data base. This is described in Section 2.2. The GPRIME preprocessor developed by David W. Taylor Naval Research and Development Laboratory outputs data in NASTRAN formats. These data may be read into the STAGING data base with the conversion routines described in Section 6.3.1.

1.5 Getting Started

The following dialogue is used to initiate the execution of STAGING in INTERCOM. The underlined text is typed by the user.

PLEASE LOGIN - Login,user ID,Password,TID
COMMAND - attach,procfil,id=STAGING3
COMMAND - begin,STAGING

This dialogue is installation specific and may vary from site to site. In addition, there are variations to this dialogue which will accomplish the same function. After the user has initiated this sequence, STAGING will begin operating.

1.5.1 Identifying the Terminal Type

The first screen message to appear requests the type of terminal being used. (See Figure 1.6). The user identifies his terminal type from the list presented and types the corresponding letter followed by a carriage return.

1.5.2 Specifying a Data Base

Figure 1.7 illustrates the screen request for the user to enter a data base. The user may enter a permanent file name (PFN) of up to 40 characters, the file identifier (ID), a specific cycle number (CYCLE), and the device set name (SETNAME). Although the data template for the PFN is broken up into 4-10 character words, the user should type in the entire PFN in one string, followed by other items separated by commas. The only required input data is PFN and ID. For example, a user could type as a

```
HELLO!  I NEED TO KNOW YOUR TERMINAL TYPE.


     ENTER A FOR A 4010.

        ENTER B FOR A 4012.

           ENTER C FOR A 4014 WITH ENHANCED GRAPHICS.

              ENTER D FOR A 4014 WITHOUT ENHANCED GRAPHICS.
```

FIGURE 1.6. SELECTING THE TERMINAL TYPE

```
A. GLOBAL
B. REDRAW
C. ERASE

 ENTER DATA BASE FILE.
   IF NEW FILE OR
 ATTACHED AS TAPEO.
    TYPE SPACE.


A) PF NAME (1)=
B)         (2)=
C)         (3)=
D)         (4)=
E) ID         =
F) PASSWORD   =
G) CYCLE      =
```

FIGURE 1.7. TYPE-IN TEMPLATE FOR DATA BASE IDENTIFICATION

minimum MYDATABASE, ID = MYNAME or alternatively A = MYDATABASE, E = MYNAME, G = 1.

If the system cannot locate the user specified data base, it erases the screen, prints an error message, and requests the user to re-enter the data. The user then retypes the correct input.

If the user does not have a data base, he enters a space and a carriage return. STAGING will allow the user to use the PREPROCESSOR to create a data base.

1.5.3 Selecting a Major Module

After specifying a data base, the system requests the user to select a major module (See Figure 1.2). STAGING consists of four major modules: GLOBAL, PREPROCESSORS, DISPLAY and EDIT, and POSTPROCESSORS. The GLOBAL module contains general purpose utilities and graphics manipulation functions. The PREPROCESSOR module is used to generate new data base items. The DISPLAY and EDIT module provides interactive capabilities for displaying data base items and for making corrections to those data base items. The POSTPROCESSOR allows easy generation of additional engineering information from analysis program outputs and effective display of analysis results.

1.6 Error Recovery

During the operation of STAGING, it is possible for events or actions to result in an error condition. Except for catastrophic host computer errors, STAGING attempts to recover from the error condition and return to a safe operating state. Three types of errors can occur:

1. Invalid operation or data
2. Arithmetic or system error
3. Host computer failure

Error TYPE 1 results from user actions which STAGING recognizes as invalid (for example, bad data input values), or which are beyond the capacity of

a given function (such as, too many data points for X-Y plots). STAGING traps these errors before they create an inconsistent or erroneous state. In general, an error message is printed and the user is asked to repeat the action or to attempt some corrective action. Table 1.2 contains a list of all TYPE 1 errors, the functions which report such errors, and recommended corrective actions.

Error TYPE 2 occurs when STAGING has not trapped an invalid operation or invalid data. In this case, the computer system traps the error and reports it to STAGING. STAGING prints a message similar to:

```
    E R R O R  R E C O V E R E D
ARITHMETIC MODE ERROR 4 AT 32756
DO YOU WISH TO CONTINUE(TYPE Y OR N)
```

Normally, the user reponds with YES or Y. This causes the screen to be erased and the menus redrawn. In general, this recovery deactivates the picture. In some such cases, STAGING may be requesting input data, but the data input box does not reappear. The user should select the RETURN button to continue. Even if the user desires to terminate the STAGING session, it is advisable to procede through a normal termination which insures some level of cleanup. If a YES response results in another TYPE 2 error, the user should respond with NO or N. ALL TYPE 2 errors should be reported to STAGING systems staff.

Error TYPE 3 means that the host computer has discovered a failure, usually unrelated to the operation of STAGING. Recovery from these failures is not guaranteed and is installation dependent. If recovery is possible, the user restarts after the host is made available. The procedures described in Section 1.2.3 are repeated and the host computer prints the following message:

```
WAITING FOR INPUT
```

The user should respond with a user abort (that is, %A, carriage return). This causes a TYPE 2 error:

TABLE 1.2. STAGING TYPE 1 ERROR MESSAGES

ERROR MESSAGE	ACTION CAUSING ERROR	ACTION USER SHOULD TAKE
BAD TYPE-IN VALUES. REENTER OR RETURN.	WHEREVER INPUT DATA ARE REQUESTED BY SYSTEM	REENTER CORRECT DATA.
CANNOT LOCATE THIS DATA BASE; TRY AGAIN	TYPE-IN PERMANENT FILE NAME AND ID.	a) REENTER CORRECT FILE NAME IDENTIFICATION. b) ENTER BLANK, PICK GLOBAL AND STOP. THEN CHECK FILE NAME AND ID.
EYE SAME AS LOOK AT. REENTER.	"ENTER NEW VIEW PARAMETERS. RETURN." 4. PERSPECTIVE	REENTER CORRECT (XEYE, YEYE, ZEYE)
ILLEGAL KEY CHARACTER OR WORD. REENTER	WHEREVER INPUT DATA ARE REQUESTED	TYPE-IN CORRECT DATA.
MAXIMUM NO. OF POINTS EXCEEDED.	X-Y PLOTS ACTIVATE DATA CONTOUR PLOTS ACTIVATE DATA	a) REQUEST PLOT OF ACTIVE DATA b) RETURN, REDUCE THE ACTIVATED SAMPLE SIZE AND REPLOT
MAXIMUM OF 10 CURVES EXCEEDED - ERASE	X-Y PLOTS "CHOOSE [TABLE NAME] FOR X-AXIS" X-AXIS "CHOOSE [TABLE NAME] FOR Y-AXIS Y-AXIS	a) PLOT ONLY 10 CURVES b) RETURN, REACTIVATE DATA TO PLOT NEW CURVES
MORE THAN ONE ACTIVE. CAN ONLY DO ONE	WHEN ADDING SUBSTRUCTURES, ELEMENTS, OR NODES WHEN DELETING ELEMENTS	DEACTIVATE ALL BUT ONE OF THE INDICATED ITEMS.
NAME ALREADY IN DATA BASE. REENTER.	WHEN CREATING A NEW BEAD	ENTER NEW NAME.

TABLE 1.2. STAGING TYPE 1 ERROR MESSAGES
(Continued)

ERROR MESSAGE	ACTION CAUSING ERROR	ACTION USER SHOULD TAKE
NO ACTIVE ATTRIBUTES. CONTINUE.	"DISPLAY ATTRIBUTES" "X-Y PLOT TABLES"	a) ACTIVATE DESIRED ATTRIBUTES b) CHECK TO SEE IF ATTRIBUTES EXIST
NO ACTIVE BEAD FOR TYPE. CREATE IN EDIT MODE.	WHEN ACTIVATING FOR DISPLAY OR DISPLAYING ALL ELEMENTS AND NODES IN DATA BASE. ALSO ON X-Y PLOTS FOR ALL STRUCTURES, ELEMENTS, AND NODES IN DATA BASE	RETURN TO EDIT MODE AND CREATE APPROPRIATE ITEMS.
NO POINTERS ACTIVE. CONTINUE.	"CREATE SUBSTRUCTURES BY NAME"	ACTIVATE ELEMENTS OR NODES BY DISPLAYING THEM. THESE ITEMS CAN THEN BE INCLUDED IN SUBSTRUCTURE.
NONE ACTIVE FOR THIS TYPE. ACTIVATE WITH DISPLAY.	"SHOW HOW ELEMENTS ARE TO BE DISPLAYED". "PICK ATTRIBUTE, THEN RETURN" "BY ATTRIBUTE RANGE" "DISPLAY ACTIVE STRUCTURES BY NAME"	ACTIVATE APPROPRIATE LEVEL BY DISPLAYING OR ACTIVATING STRUCTURES AND SUBSTRUCTURES FOR DISPLAY.
NOT ENOUGH ROOM FOR SPLIT-SCREEN.	SPLIT SCREEN	LIMIT AMOUNT OF ITEMS DISPLAYED AND THEN SPLIT SCREEN
NUMBERS BELOW NOT IN DATA BASE. REENTER.	WHENEVER A TYPE-IN VALUE CONSISTS OF A NUMBER OR A RANGE OF NUMBERS.	TYPE IN NUMBERS THAT ARE IN DATA BASE.
X AND/OR Y UNDEFINED	X-Y PLOT "X-AXIS" "Y-AXIS"	DEFINE ITEMS TO BE PLOTTED ON X OR Y AXIS.

```
E R R O R   R E C O V E R E D
USER ABORT AT 16584
DO YOU WISH TO CONTINUE (TYPE Y OR N)
```

The proper response is YES (Y). If TYPE 3 errors are recovered, the system is probably in a safe operating state. The user must be aware of local host computer recovery procedures in order to determine whether TYPE 3 errors are recoverable, and under what conditions.

CHAPTER 2

GENERATING STAGING DATA BASES

STAGING provides several means for creating a finite element model and generating a STAGING Data Base. This chapter describes the capabilities of the PREPROCESSOR module and data conversion utility subroutines. Chapter 5 discusses methods of adding or changing items in an existing data base. Chapter 6 documents the use of data conversion programs for creating a STAGING Data Base from analysis program inputs.

The GPRIME geometry and mesh generation capability can be used to define generalized curves and surfaces and automatically generate a finite element mesh. The DRAFTING function is a two dimensional sketchpad which is used to draw finite element models on the screen. To support the generation of a model's material properties, the ASSIGN MATERIAL PROPERTIES function creates material properties tables and stores them in the model data base. Each of the above functions is located under the PREPROCESSOR module.

STAGING has a set of seven utility subroutines which supports the generation of data bases from externally created geometry and mesh data. These utility subroutines have been developed to aid the user in writing data conversion programs. Six data conversion programs have already been written and are available for converting analysis program input data into a STAGING Data Base for NASTRAN, AXISOL, DOASIS, HONDO, FASTOP, and ADINA. (See Section 6.3)

2.1 GPRIME

GPRIME allows a user with a rudimentary background in the concepts of geometry to easily define any general curve or surface. A large repertoire of geometric constructions and manipulative abilities are available to the user. Not only can the user define classical geometric items such as cones and spheres, but he is able to define any general curve or surface not otherwise definable through a digitizing process. The user may determine the curve of intersection between two arbitrary surfaces, and the point of intersection of two arbitrary curves, of a curve and a surface, and of three surfaces.

GPRIME attempts wherever possible to provide several different methods of describing a particular geometric construction. An example is the CYLINDER definition form, where the user may define a cylinder by specifying its axis of symmetry and either a point on one of the base circumferences or a radius.

GPRIME allows the user to quickly translate a drawing or blueprint into a mathematical model. Alternatively, the user may create his structure using GPRIME directly. The user has the ability to intermix digitized data, prestored language statements and interactive commands and definitions. The user may also create groups of language statements called macros and execute them wherever desirable. The user's manual for GPRIME will be available as a separate document.

The GPRIME geometry generation package was developed by the David W. Taylor Naval Ship Research and Development Center as a preprocessor for STAGING. At the present time, it operates as a stand-alone module. GPRIME includes the subprograms for generating shell elements in either STAGING data base formats or NASTRAN formats. A three dimensional mesh generation system (SOLIDGEN) has been developed for creating solid three dimensional models. SOLIDGEN operates in a batch mode and creates node and element tables in NASTRAN formats. Thus SOLIDGEN created models may be stored in the STAGING data base through the NASTRAN conversion routine described in Section 6.3.1.

Documentation of these generation packages is currently under development.

2.2 DRAFTING

The DRAFTING function is a two dimensional sketchpad which is used to draw or draft simple finite element models. It may be used to create a new data base or to add a limited number of nodes and elements to an existing data base. Figure 2.1 shows the DRAFTING screen displayed by the system. The default space used by DRAFTING has a minimum X value of -1.05 and a maximum X value of +1.05; the Y values are the same.

2.2.1 GRID ON/OFF

GRID ON/OFF turns the DRAFTING reference grid on and off. If the grid is displayed and GRID ON/OFF is selected, the grid will be turned off. The system does not automatically erase the screen when the grid is turned off. The user can pick REDRAW (See Section 1.3.2) and the system will erase the screen and redraw only nodes and elements. If the grid is not displayed and GRID ON/OFF is selected, the grid is automatically drawn.

After the user has constructed a model or parts of a model, GRID ON/OFF can be used to remove the grid lines in order to more clearly see the model.

2.2.2 CHANGE PARAMETERS

CHANGE PARAMETERS allows the user to define the distance between the lines in the DRAFTING reference grid. The default distance is 64 units; this represents approximately .04 units in the DRAFTING space. The users changes the distance by entering a new value from the keyboard. The initial data input template is:

A) GRID UNITS = 64

Data values less than 64 create grids which are very fine and may be hard to use. A data value of 300 creates a reference grid with lines which are (visually) approximately one inch apart. A data value of 295 creates lines approximately .2 DRAFTING space units apart. Figure 2.2 shows the effect of changing the grid unit parameter.

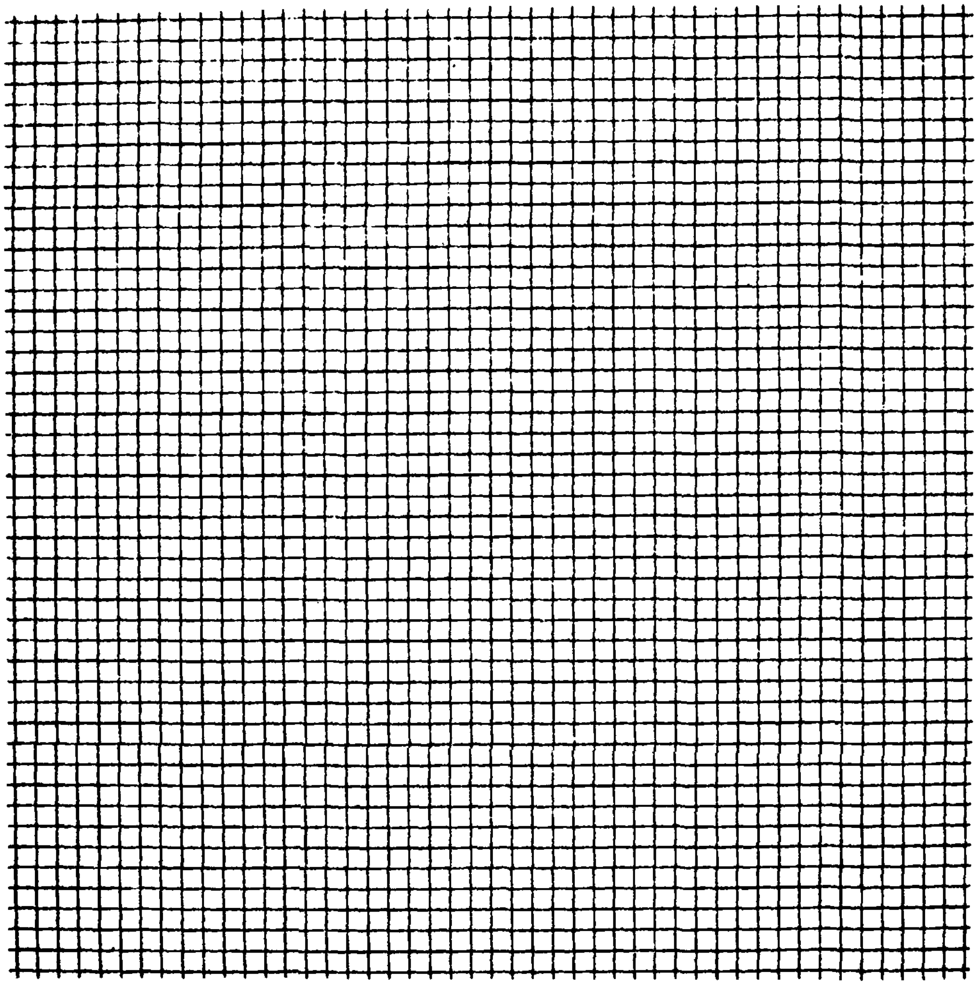


FIGURE 2.1. THE DRAFTING GRID

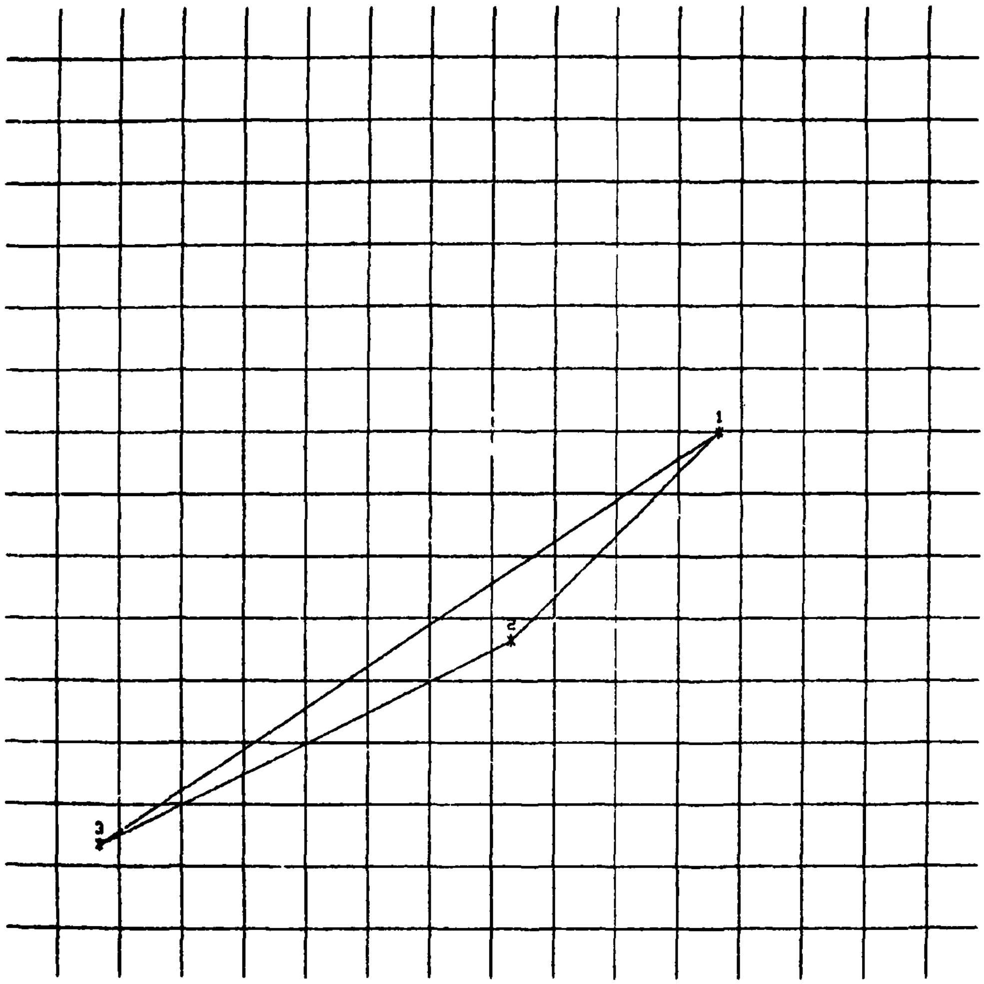


FIGURE 2.2. CHANGING GRID SCALE

2.2.3 SET LINE MODE

SET LINE MODE indicates that lines are drawn between nodes as they are defined. Furthermore, the line between every two nodes is considered an element (of type ROD) and is stored in the data base.

2.2.4 SET POINT MODE

SET POINT MODE indicates that nodes are to be drawn disconnected as they are identified. This is the default mode when DRAFTING is initialized. It may be changed by SET LINE MODE (See Section 2.2.3). After nodes are created, element connectivity can be established using CREATE ELEMENTS (See Section 2.2.6)

2.2.5 CREATE NODE

CREATE NODE allows the user to identify locations within the 2D DRAFTING space and create nodes with those coordinates. When CREATE NODE is selected, the crosshairs reappear immediately. They are then positioned to the desired location and any keyboard character entered. The system creates a new node and stores it in the data base. The new node is drawn on the screen as an asterisk (*) and its number displayed. New nodes are numbered consecutively, starting with one higher than the highest node number currently in the data base. (An efficient procedure for creating nodes is to position crosshairs at the desired nodal position, type the letter designation for "CREATE NODE" and then hit the same character to make the node pick at the crosshair location.

If SET LINE MODE has been selected, a rod element is also created and stored in the data base. The element is displayed on the screen as a line between the new node and the previously created node.

2.2.6 CREATE ELEMENTS

CREATE ELEMENTS allows the user to define elements and store them in the data base. Using the crosshairs in this mode, the user

sequentially identifies the nodes in the order they are to be connected in the element. After the nodes of an element have been identified, the user selects the CONNECT menu button.

2.2.7 DELETE ELEMENTS

DELETE ELEMENTS removes elements from the data base. The elements to be deleted are identified with the crosshairs. The screen is not erased as elements are deleted. The REDRAW button can be selected to obtain a current picture.

2.2.8 DELETE NODES

DELETE NODES removes nodes from the data base. The nodes to be deleted are identified with the crosshairs. Furthermore, references to these nodes are deleted from any element to which they are connected. When DELETE NODES is selected, the screen is not automatically erased, although the modified element is drawn. Selecting the REDRAW button will make the picture current.

2.3 ASSIGNING MATERIAL PROPERTIES

The material properties information for finite element models is stored in the STAGING Data Base as follows:

1. Each element in the model has an the attribute MATERIAL. This is the same material which analysis programs use to identify material type. It is an integer number (1 to 99999).
2. Tables of actual material properties are stored in Level 5 within the STAGING Data Base. The number of tables is defined by the user. Each table has a unique identification number (as part of the table name) corresponding to the material identification number for elements.

2.3.1 Material Properties Input to STAGING

The materials property information can be added to the STAGING Data Base in two ways:

1. Conversion Routines (See Section 2.4).
2. The Preprocessor module of STAGING.

2.3.1.1 Conversion Routine Approach. When the user is constructing an initial data base by using Conversion Program 1 (see Section 2.4) the material properties can also be assigned at that time. This can be done as follows:

1. Define the MATERIAL attribute for elements as being active.
2. In the call to subroutine CNELEM, (Section 2.4.3) the attribute array ATT should contain the material identification number (MID) of the element in the appropriate location.
3. Create a material property table using subroutine CNTABL. (Section 2.4.6) The call to CNTABL is:

   Call CNTABL (NAME, ARRAY, 44, 7H(F10.0)) where NAME, the material property table name, is generated before the call to CNTABLE with the following FORTRAN statements:

   ENCODE (40, 101, NAME) MID

   101 FORMAT (30H MATERIAL PROPERTIES TABLE, I10)

   and the contents of ARRAY (44 words long) is shown in Table 2.1 is also supplied by the user.

Note that the number of values in ARRAY depends on the number of temperatures at which the user supplies properties. If he has properties at 5 temperatures (ARRAY (1) = 5) then he can supply values for temperatures T_1 to T_5 and leave the remaining entries in ARRAY undefined.

2.3.1.2 Preprocessor Module Approach. This approach is to select element material properties using the Preprocessor Module. The command tree for assigning material property is accessed using the STAGING preprocessor button. Through the use of this command tree, the user performs two functions.

TABLE 2.1. CONTENTS OF MATERIAL PROPERTY TABLE IN STAGING DATA BASE

Word	Contents
1	Number of temperature points
2	Mass density
3	Unused
4	Temperature, T_1
5	Young's Modulus at T_1
6	Poisson's Ratio at T_1
7	Coefficient of Thermal Expansion at T_1
8 to 40	Properties of T_2 to T_9
41	Temperature at T_{10}
42	Young's Modulus at T_{10}
43	Coefficient of Thermal Expansion at T_{10}
44	Poisson's Ratio at T_{10}

1. Selecting elements
2. Selecting a material from the STAGING MATERIAL PROPERTY LIBRARY (MPDB) or typing in material properties.

STAGING allows selection of elements from the user's data base for the following cases: -

1. By Structure - a. All elements
   b. Elements of a given type
   c. Range of element numbers
2. By Substructure (for Active SS) - a. All elements
   b. Elements of a given type
   c. Range of element numbers.

After selecting the elements and a material the user is instructed to provide an indentification number for the selected materials. The STAGING system now updates two portions of the data base.

1. The MATERIAL attribute - if the attribute was previously defined, the value is changed. However, if no MATERIAL attribute is present, a new attribute is created and a value is assigned to selected elements.
2. A table is created in the STAGING Data Base. The contents and the format of this table is identical to the table described above in Section 2.3.1.1..

2.3.2 Material Property Data Base

The materials included in STAGING reside on Materials Properties Data Base (MPDB). Table 2.2 shows a list of materials available to the user. The modifications to MPDB are the responsibility of the STAGING system maintenance personnel. The STAGING system manual describes the format of the MPDB and procedures to add or delete materials and their properties.

TABLE 2.2. MATERIALS INCLUDED IN MPDE

Number	Generic	Specific	Temp Range °F
	STEEL		
1		AM-350	80. - 1000.
2		SCR-MO-V	80. - 1200.
3		AISI 301	80. - 1200.
4		A286	0. - 1200.
	ALUMINUM		
5		6061-T6	-200. - 600.
6		7178-T6	100. - 600.
7		2025-T6	-200. - 600.
	NICKEL		
8		INCO X750	100. - 1200.
9		RENE 41	100. - 1200.
	BERYLLIUM		
10		Beryllium	80. - 1400.
	MAGNESIUM		
11		AZ31B-0	100. - 600.
12		HK31-A	80. - 600.
	TITANIUM		
13		T1-6AL-4V	80. - 1000.

2.4 Converting Data into the Data Base

This section describes facilities available to a user for writing a program to create a STAGING Data Base from an analysis program input data file. Conversion programs for six finite element analysis programs are available with STAGING. These include NASTRAN, FASTOP, AXISOL, DOASIS, HONDO, and ADINA. Section 6.3 of this manual describes the use of these conversion programs.

A set of seven utility subroutines have been developed to aid in creating STAGING data bases. These subroutines initialize the data base, define items and attributes on each level, and close the data base. The initialization routine (CNINIT) must be executed before any other routines are called. The termination routine (CNTERM) must be the last routine called. The other subroutines (CNODE, CNELEM, CNSUB, CNSTR, and CNTABL) may be called in any order.

Each item in the data base is assigned a name or number, and has an associated list of attributes that describe it. The data base contains five levels of information: structures, substructures, elements, nodes, and tables. Nodes are grouped to make elements, elements are grouped to make substructures, and substructures are grouped to make structures. Tables are used to store tabular data or alphanumeric information associated with a data base. (Additional detail on the STAGING Data Base is given in Section 1.4). When a user creates a data base using these utility subroutines, he must supply both the grouping and the attribute list. It is also possible to supply attribute values during data base creation, or later using editing features (See Chapter 5).

The following sections describe each of the seven utility subroutines and their use. The last section provides an example of a program to create a STAGING Data Base.

2.4.1 CNINIT

Subroutine CNINIT initializes the data base and defines the attribute lists for nodes and elements. It is used as follows:

CALL CNINIT (IATNOD, NATNOD, IATELM, NATELM)

where: IATNOD is the node attribute array
NATNOD is the number of node attributes
IATELM is the element attribute array
NATELM is the number of element attributes.

Both NATNOD and NATELM are positive integer values. NATNOD must be less than or equal to 28; NATELM must be less than or equal to 41; IATNOD and IATELM are arrays of ten character variables. These ten character variables specify the names of attributes which are associated with nodes or elements. Table 2.3 lists the node attributes available in STAGING. Table 2.4 lists the element attributes in the system.

The first seven (7) characters of each attribute name are unique. A subscripting capability is provided to allow any node or element attribute to assume multiple values. For example, a time dependent analysis may define five (5) different normal stress values (e.g. X N STRESS). These subscripted normal stresses are identified by specifying a ten character field in which the first seven characters are the same (e.g. X N STR) and the remaining three characters (digits) define subscripts. For example:

IATELM (1) = 'X N STR 1 '
IATELM (2) = 'X N STR 2 '
IATELM (3) = 'X N STR 3 '
IATELM (4) = 'X N STR 4 '
IATELM (5) = 'X N STR 5 '

(Note: The attribute names 'X N STRESS', 'X N STR 0 ', and X N STR 1 ' all refer to the same attribute.)

When CNINIT is called, it reserves space in the data base for each attribute specified. New attributes may be added at any time (See Chapter 5.) However, it is more efficient to specify as many attributes as possible during data base creation even though the attribute values may be added later using other features.

The order in which attribute names are specified in IATNOD and IATELM fixes their relative positions for subsequent calls to CNNODE (See

TABLE 2.3 STAGING NODE ATTRIBUTES

No.	Full Name	Attributes	Abbreviation	Values
1	X Coordinate	X-CORD	X	any[1]
2	Y Coordinate	Y-CORD	Y	any[1]
3	Z Coordinate	Z-CORD	Z	any[1]
4	Coordinate System	CORD-SYSTM	COS	1=Cartesian 2=Polar(2D)Cylindrical(3D) 3=Spherical(3D Only)
5	X Force	X-FORCE	XFO	any
6	Y Force	Y-FORCE	YFO	any
7	Z Force	Z-FORCE	ZFO	any
8	X Moment	X-MOMENT	XMO	any
9	Y Moment	Y-MOMENT	YMO	any
10	Z Moment	Z-MOMENT	ZMO	any
11	Force Coordinate System	FORCE-CORD	FCO	Local Cord Sys. ID
12	Balance Weight	BAL WT	BWT	any
13	X Rotation	X-ROTATION	XRO	any
14	Y Rotation	Y-ROTATION	YRO	any
15	Z Rotation	Z-ROTATION	ZRO	any
16	Displacement Coordinate System	DISP-COORD	DCO	Local Coord Sys. ID
17	Temperature	TEMPERATUR	TEM	any

TABLE 2.3 (Continued)

No.	Full Name	Attributes	Abbreviation	Values
18	Concentrated Mass	CONC-MASS	CMS	any
19	Pressure	PRESSURE	PRE	any
20	X Displacement	X DISP	XDS	any[2]
21	Y Displacement	Y DISP	YDS	any[2]
22	Z Displacement	Z DISP	ZDS	any[2]
23	X Mode Shape	X MODE	XMD	any[2]
24	Y Mode Shape	Y MODE	YMD	any[2]
25	Z Mode Shape	Z MODE	ZMD	any[2]
26	X Load	X LOAD	XLD	any[2]
27	Y Load	Y LOAD	YLD	any[2]
28	Z Load	Z LOAD	ZLD	any[2]

[1] Interpretation of X, Y, Z varies according to Attribute 4. If CORD-SYSTM. is 1.0 (Cartesian), X, Y, Z are coordinates in space. If CORD-SYSTM is 2.0 and only X and Y are provided (Polar), Attribute 1 is R and Attribute 2 θ measured in radians. If X, Y, Z are provided (cylindrical), Attribute 3 is Z as in the Cartesian case. If X, Y, Z are provided and CORD-SYSTM is 3.0 (spherical), Attribute 1 is R, 2, θ (radians) and 3, ϕ (radians).

[2] The displacement factors are delta displacements from the node Point X, Y, A and must be provided in the same coordinate system. The delta values can refer to up to ten mode shapes, load conditions, or time steps. Only one of these conditions (mode shapes, load conditions, or time steps) can be active at any one time in the data base.

TABLE 2.4 STAGING ELEMENT ATTRIBUTES

No.	Full Name	Attribute Name	Abbreviation
1	Type	TYPE *	TYP
2	Material Identifier	MATERIAL	MAT
3	Area Cross-Section	AREA-CRSST	CSA
4	Area Moment X**Direction	X AREA MOM	XAM
5	Area Moment Y Direction	Y AREA MOM	YAM
6	Area Moment Z Direction	Z AREA MOM	ZAM
7	Torsional Constant	TORSIONAL	TOR
8	Mass/Length	MASS/LENGT	MPL
9	Membrane Thickness	MEM THICK	MTH
10	Mass/Area	MASS/AREA	MPA
11	Flexual Thickness	FLEX THICK	FTH
12	Material Property A	MAT-PROP-A	PRA
13	Pressure	PRESSURE	PRE
14	Temperature	TEMPERATUR	TEM
15	Critical Load	CRIT LOAD	CLD
16	Design Criterion	DES CRIT	DCR
17	Construction Code	CONSTRCODE	CCD
18	Geometry Class	GEOMCLASS	GCL
19	Geometry Sub-Class	SGEOMCLASS	SGC
20	Angle Between Prop-Axes & Side I-T	BETA	BET
21	Tension Allowable Stress	TEN ALWSTR	TAL
22	Compression Allowable Stress	CMP ALWSTR	SAL
23	Shear Allowable Stress	SHR ALWSTR	SAL
24	Minimum Size	MIN SIZE	MIN
25	Maximum Size	MAX SIZE	MAX
26	Allowable Class	ALOWCLASS	ALC
27	Allowable Sub-Class	SALOWCLASS	SCN
28	Average Stress Concentration Ratio	STRCNSTR	STC
29	Original Thickness	ORIG THICK	OTH
30	Excluded Element	EXCLUD ELM	EXE
31	Non-Optimum Weight Factor	NOPTWTFAC	NPW
32	Normal Stress X (Centroid)	X N STRESS	XNS
33	Normal Stress Y (Centroid)	Y N STRESS	YNS
34	Normal Stress Z (Centroid)	Z N STRESS	ZNS
35	Shear Stress X (Centroid)	X S STRESS	XSS
36	Shear Stress Y (Centroid)	Y S STRESS	YSS
37	Shear Stress Z (Centroid)	Z S STRESS	ZSS
38	Maximum Principal Stress	MAX STRESS	MXS
39	Intermediate Principal Stress	INT STRESS	INS
40	Minimum Principal Stress	MIN STRESS	MNS
41	Equivalent Stress	EQU STRESS	EQS

*See Table 2.5.3.

**XYZ - GLOBAL Cartesian Coordinate System

Section 2.4.2) and CNELEM (See Section 2.4.3). If attribute values are supplied using CNNODE or CNELEM, the order of the values must correspond to the order of the names.

2.4.2 CNNODE

Subroutine CNNODE creates a node and stores its attribute values in the data base. It is used as follows:

```
CALL CNNODE (NUM, ATT, NATT)
```

Where: NUM is the node number
ATT is the attribute value array
NATT is the number of attribute values.

Both NUM and NATT are positive integer values. All values of NUM must be unique within a given data base. NATT must be less than or equal to NATNOD (See Section 2.4.1). ATT normally contains floating point (or real) values.

2.4.3 CNELEM

This subroutine creates elements, defines node linkage, and stores attribute values in the data base. It is used as follows:

```
CALL CNELEM (NUM, NODE, NNODE, ATT, NATT)
```

Where: NUM is the element number
NODE is an array containing the node number of nodes belonging to this element
NNODE is the number of numbers in array NODE
ATT is an attribute value array
NATT is the number of attribute values.

NUM, NNODE, and NATT are all integer values. All values for NUM must be unique in a given data base. NATT must be less than or equal to NATELM

(See Section 2.4.1). NODE is an array of integer values. ATT is an array of several data types; it may include floating point values, integer values, or character values.

When elements are created with CNELEM, the attribute TYPE must be specified. Table 2.5 lists the element types (that is, attribute values for TYPE) available in STAGING.

Typically, adjacent elements contains identical nodes. It is not necessary to have unique nodes in the data base for elements which use the same nodes.

2.4.4 CNSUB

Subroutine CNSUB creates a substructure and stores its element linkage into the data base. It is used as follows:

```
CALL CNSUB (NAME, IELEM, NELEM)
```

Where: NAME is the substructure name
IELEM is an array of element numbers of elements which belong to this substructure.
NELEM is the number of elements.

NAME is an array of forty (40) characters (or four words), and must be unique within a given data base. Both IELEM and NELEM are integer values.

2.4.5 CNSTR

This subroutine creates a structure and stores the substructure linkage in the data base. It is used as follows:

```
CALL CNSTR (NAME, SUBS, NSUB)
```

Where: NAME is the structure name
SUBS is an array of names of substructures that belong to this structure
NSUB is the number of substructures.

TABLE 2.5 STAGING ELEMENT TYPES

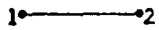
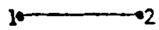
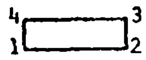
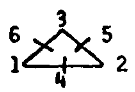
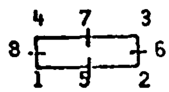
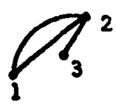
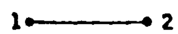
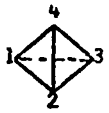
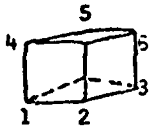
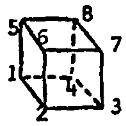
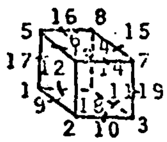
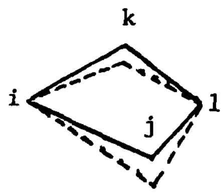
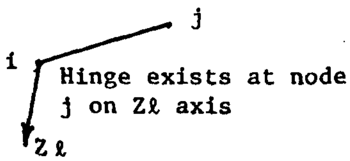
TYPE NO.		FULL NAME	SHORTHAND NAME
1		Rod	ROD
2		Straight Beam	STR BEAM
3		Membrane Triangle	MEM TRIA
4		Membrane Quadrilateral	MEM QUAD
5	Same as 3	Plate Triangle	PLATE TRIA
6	Same as 4	Plate Quadrilateral	PLATE QUAD
7	Same as 3	General Triangle	GEN TRIA
8	Same as 4	General Quadrilateral	GEN QUAD
9	Same as 3	Ring Triangle	RING TRIA
10	Same as 4	Ring Quadrilateral	RING QUAD
11	Same as 4	Shear Panel	SHER PANEL
12	Same as 4	Twist Panel	TWIS PANEL
13		General Triangle(2)	GEN TRIA2
14		General Quadrilateral(2)	GEN QUAD2
15	Same as 13	Ring Triangle(2)	RING TRIA2
16	Same as 14	Ring Quadrilateral(2)	RING QUAD2
17		Curved Beam	CURVE BEAM
18	Same as 17	Ring Shell	RING SHELL
19		Ring Conical	RING CNICL

TABLE 2.5. STAGING ELEMENT TYPES
(Continued)

TYPE NO.		FULL NAME	SHORTHAND NAME
20		Tetrahedral Solid	TETR SOLID
21		Wedge Solid	WEDG SOLID
22		Hexahedral Solid	HEXA SOLID
23		20 Node Brick	BRICK20
24	1—AS—2	Axial Spring	AXIAL SPRG
25	1—TS—2	Torsional Spring	TORSN SPRG
26	1—M—2	Mass	MASS
27	1—D—2	Damper	DAMPER
28	 warped quadrilateral	Warped Quadrilateral	WARP QUAD
29		Hinged Beam	HINGE BEAM

NAME is an array of forty (40) characters, and must be unique within a given data base. SUBS is an array of forty (40) character substructure names. NSUB is an integer value. In practice, a data base only contains one structure. However CNSTR may be called several times, if the user wishes to create multiple structures in a single data base.

2.4.6 CNTABL

Subroutine CNTABL supports the storage of tabular data within the data base. It is used as follows:

```
CALL CNTABL (NAME, ARRAY, NR, NC, ND, FORMAT)
```

Where: NAME is the name of the table
ARRAY is an array containing data to be stored
NR is the row dimension of ARRAY
NC is the column dimension of ARRAY
ND is the depth dimension of ARRAY
FORMAT is data format.

NAME is an array of forty (40) characters, and is used to uniquely identify the table within the data base. NR, NC, and ND are all integer values. FORMAT is an array of up to thirty (30) characters of FORTRAN format specification.

CNTABL is used to store data not directly associated with elements or nodes (for example, analysis program options). STAGING provides limited capability for graphing data stored in tables (See Chapter 6).

2.4.7 CNTERM

This subroutine closes the data base and is the last subroutine to be called. It is used as follows:

```
CALL CNTERM.
```

2.4.8 Sample Program

This section describes a simple conversion program to create a STAGING data base from an input file containing nodes and element information.

Figure 3 shows a sample conversion program. Input to the program consists of a file (TAPE1) consisting of node and element information. The output consists of a data base file (TAPE0). The sample program is commented to describe each line of coding.

The user can execute the sample program using the following job control cards.

a. Compile the Sample Program.

b. Attach the input data as TAPE1. (Figure 2.4)

c. Attach the library of STAGING conversion programs:
ATTACH,LIB,SUPPORTLIB,ID=STAGING3.
LIBRARY,LIB.

d. LGO. Execute the program.

e. Catalog TAPE0,MYDATABASE,ID=MYID.

3

```
      PROGRAM TESTCON(INPUT,OUTPUT,TAPE1,TAPE77)
      DIMNESION IATNOD(3),IATELM(2),ATT(3),IX(4)
C     SAMPLE CONVERSION PROGRAM-----------
C             TO CREATE A STAGING DATA BASE
C             FROM TAPE1(NODES AND ELEMENTS)
C
C     DEFINE NUMBER OF NODE AND ELEMENT ATTRIBUTES
C
      NATNOD=3
      NATELM=2
C     DEFINE TYPE OF NODE AND ELEMENT ATTRIBUTES
      IATNOD(1)=10HX-CORD
      IATNOD(2)=10HY-CORD
      IATNOD(3)=10HZ-CORD
      IATELM(1)=10HTYPE
      IATELM(2)=10HMATERIAL
C     INITIALIZE STAGING DATA BASE
      CALL CNINIT(IATNOD,NATNOD,IATELM,NATELM)
      REWIND 1
      READ(1,100)NUMNP,NUMEL
      DO 1 I=1,NUMNP
      READ(1,101) NODE,ATT(1),ATT(2),ATT(3)
C     DATA BASE CALL
      CALL CNNODE(I,ATT,NATNOD)
    1 CONTINUE
C     READ ELEMENT CONNECTIVITY AND ADD TO STAGING DB
C     TYPE OF ELEMENT IN THIS PROGRAM IS A QUADRILATERAL
      ATT(1)=10HGEN QUAD
      DO 2 I=1,NUMEL
      READ(1,100) IEL,(IX(J),J=1,4),MAT
      ATT(2)=MAT
C     DATA BASE CALL
      CALL CNELEM(IEL,IX,4,ATT,NATELM)
    2 CONTINUE
C     TERMINATE THE DATA BASE
      CALL CNTERM
  100 FORMAT(5I5)
  101 FORMAT(I5,3F10.0)
      STOP
      END
```

FIGURE 2.3. A SAMPLE DATA CONVERSION ROUTINE

10	4				
1	0.0		0.0		0.0
2	1.0		0.0		0.0
3	2.0		0.0		0.0
4	3.0		0.0		0.0
5	4.0		0.0		0.0
6	0.0		1.0		0.0
7	1.0		1.0		0.0
8	2.0		1.0		0.0
9	3.0		1.0		0.0
10	4.0		1.0		0.0
1	1	2	7	6	1
2	2	3	8	7	1
3	3	4	9	8	1
4	4	5	10	9	1

FIGURE 2.4. INPUT DATA FOR TESTCON

2.5 Summary

STAGING has several capabilities for generating data bases. The GPRIME preprocessor is a generalized geometry generator which also has the capability to generate both plate element meshes and solid element meshes. DRAFTING is a two dimensional sketch pad which can be used to draw planar finite element meshes. Under the STAGING Preprocessor there is also the capability to assign material properties to elements using the Material Properties Data Base. Additionally, there is a set of seven utility subroutines to facilitate development of conversion programs. These create a data base from externally generated data (such as a digitizer or external mesh generator). Figure 2.5 illustrates the STAGING Preprocessor menu tree.

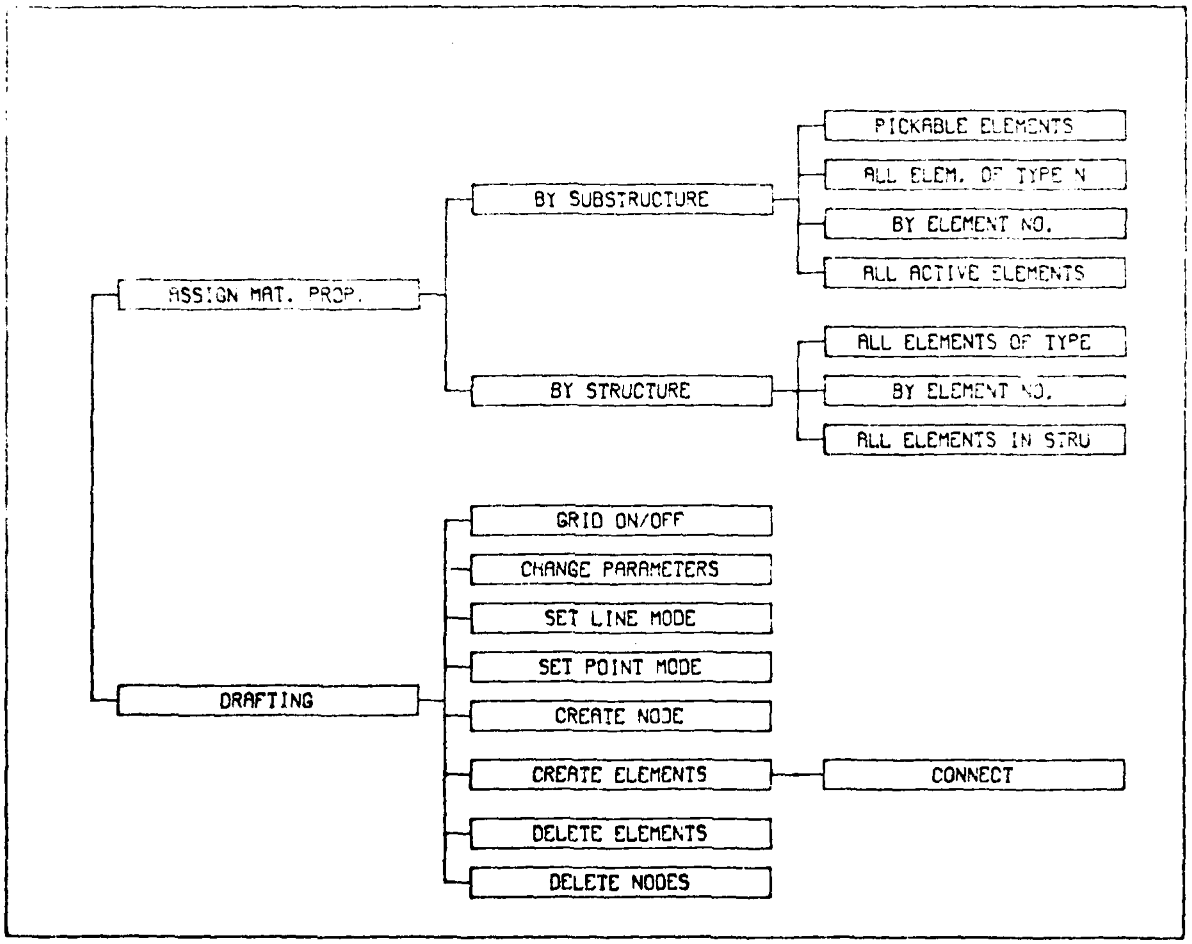


FIGURE 2.5. THE STAGING PREPROCESSOR TREE

CHAPTER 3

DISPLAYING FINITE ELEMENT MODELS

The STAGING Sytem provides several functions which allow the user to display selected portions of his model and selected data (such as attributes) about his model. This section describes these capabilites and how to use them. In conjunction with functions for manipulating the view of a model (See Chapter 4, Modifying the Picture) and functions for editing the model geometry (See Chapter 5, Editing Finite Element Models), the user can rapidly and easily verify a model **before** submitting it for analysis.

3.1 Activating for Display

The process of specifying which part of the model (including the entire model) is to be displayed is called **activating.** After an item (such as a structure, substructure, element, or node) is activated, it is considered **active.** When nodes are activated, they are automatically displayed as asterisks. When elements are activated, they are automatically displayed with solid lines connecting the appropriate nodes. If both nodes and elements are active, both the asterisks and the lines are displayed (See Figure 3.1).

If substructures or structures are activated the picture is not automatically drawn. Active substructures merely define the aggregation of elements and nodes which are to be considered. Active structures limit the collection of substructures which are to be considered. In other words, active structures or active substructures define a subset of the model to be displayed.

When a given item is activated, **only** that item is activated; the items below it are not activated. (Note that nodes do not have items below them). For example, when an element is activated, the nodes which define the element are not activated (and, therefore, the asterisks are not displayed). If a substructure is activated, and drawn, all of the elements belonging to that substructure are displayed as **a single item.** But, the elements are not activated for other functions such as displaying element attribute values.

In general, there are several different ways of identifying items to be activated. These activating functions vary slightly between the different levels of the data base. Structures can be activated by name or as all structures in the data base (See Section 3.3.1). Substructures can be activated by name or as all substructures in the data base (See Section 3.4.1). Elements can be activated by number, as all elements belonging to active substructures, as all elements with specified attribute values, or as all elements in the data base (See Section 3.5.1). Nodes can be activated by number, as all nodes belonging to active elements or substuctures, as nodes belonging to selected elements or as all nodes in the data base (See Section 3.6.1). (Note, however, that activating either

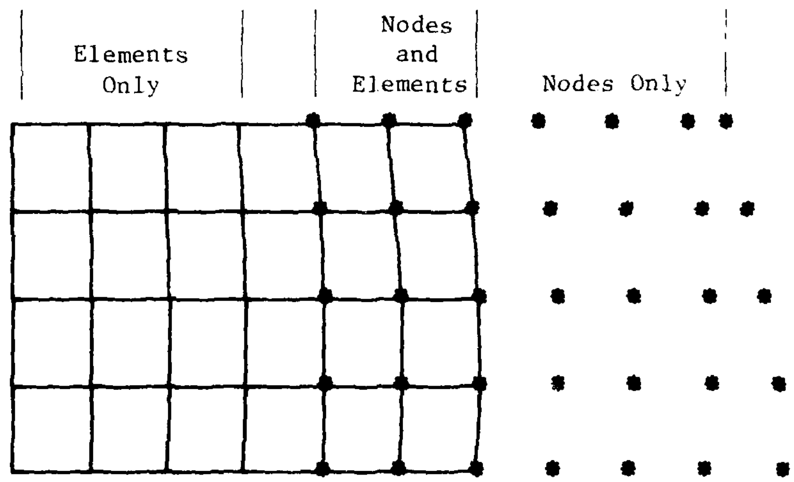


FIGURE 3.1. NODE AND ELEMENT DISPLAY FORMATS

elements or nodes is limited by active substructures, except when they are activated by number).

In addition to identifying items to be displayed, the process of activating is also used to identify items for editing (see Section 5.1), for shrinking (see Section 4.6), and for X-Y plots (see Section 7.3), and for contour plots (see Section 7.4). In each of these cases, the methods of activating may vary slightly. The general concepts and rules presented above apply to all cases.

3.2 2D MODELS and 3D MODELS

STAGING allows the user to display a model in 2D or 3D. When models are displayed in 2D, the Z coordinate is assumed to be zero. Therefore, a 3D model can be displayed in 2D. If a 2D model is assigned a zero Z coordinate, it can be displayed (and manipulated) in 3D.

3.2.1 2D MODELS specifies that the model is to be displayed in 2D. The Z coordinate is ignored even if the model does not have a Z coordinate (that is, the Z-CORD node attribute is absent).

3.2.2 3D MODELS specifies that the model is to be displayed in 3D. This means that the Z coordinate **must** be present; that is, the Z-CORD node attribute must be in the data base and have a defined value.

3.3 STRUCTURES

The STRUCTURES function indicates that the user is displaying structures. When STRUCTURES is first selected, the system displays the menu shown in Figure 3.2.a. The complete structures menu is shown in Figure 3.2.b. Each of the buttons in this menu is discussed below.

3.3.1 ACTIVATE FOR DISPLAY

The ACTIVATE FOR DISPLAY function specifies which STRUCTURES in the data base are to be activated. (See Section 3.1 for more detail on

A. GLOBAL
B. REDRAW
C. ERASE

DISPLAY OR EDIT
STRUCTURES

D. ACTIVATE FOR DISPLAY

I. CREATE

L. EDIT STRUCTURES
M. RETURN

a. First Appearance

A. GLOBAL
B. REDRAW
C. ERASE

DISPLAY OR EDIT
STRUCTURES

D. ACTIVATE FOR DISPLAY
E. DISPLAY ATTRIBUTE
F. DEACTIVATE
G. RETAIN
H. DRAW PICTURE
I. CREATE
J. DELETE FROM DATA BAS
K. MERGE
L. EDIT STRUCTURES
M. RETURN

b. Complete Menu

FIGURE 3.2. THE STRUCTURES MENU

activating.) The options for identifying structures are:

3.3.1.1 BY NAME.
3.3.1.2 SEARCH ALL IN DATA BASE.
3.3.1.3 ALL IN DATA BASE.

3.3.2 DISPLAY ATTRIBUTE

The DISPLAY ATTRIBUTE function (under STRUCTURES) allows the user to identify and display attribute values for active structures. The options for identifying active structures are:

3.3.2.1 BY CROSSHAIRS.
3.3.2.2 ALL ACTIVE.

3.3.3 DEACTIVATE

The DEACTIVATE function (under STRUCTURES) removes selected structures from the current list of active structures. The options for identifying structures to be deactivated are:

3.3.3.1 BY NAME.
3.3.3.2 BY ATTRIBUTE RANGE.
3.3.3.3 BY CROSSHAIRS.
3.3.3.4 ALL ACTIVE.

3.3.4 RETAIN

The RETAIN function (under STRUCTURES) deactivates all active structures except those selected with the crosshairs.

3.3.5 DRAW PICTURE

The DRAW PICTURE function (under STRUCTURES) displays all currently active structures. If other items, such as nodes or attributes,

are active they are also displayed. The screen is **not** automatically erased when DRAW PICTURE is selected.

3.4 SUBSTRUCTURES

The SUBSTRUCTURES functions allow the user to identify substructures in the data base to be displayed. When SUBSTRUCTURES is first selected, the system displays the menu shown in Figure 3.3.a. The complete substructures menu is shown in Figure 3.3.b.

3.4.1 ACTIVATE FOR DISPLAY

The ACTIVATE FOR DISPLAY function specifies which SUBSTRUCTURES in the data base are to be activated: (See Section 3.1 for more detail on activating). When substructures are activated, they are not automatically drawn. DRAW PICTURE (see Section 3.4.5) must be selected in order to have the substructure drawn. The options for identifying substructures are:

3.4.1.1 BY NAME.
3.4.1.2 SEARCH ALL IN DATA BASE.
3.4.1.3 ALL IN DATA BASE.

3.4.2 DISPLAY ATTRIBUTE

The DISPLAY ATTRIBUTE function (under SUBSTRUCTURES) allows the user to display attribute values for selected substructures. The options for selecting the substructures are:

3.4.2.1 BY CROSSHAIRS.
3.4.2.2 ALL ACTIVE.

3.4.3 DEACTIVATE

The DEACTIVATE function (under SUBSTRUCTURES) removes user identified substructures from the list of currently active substructures.

A. GLOBAL	A. GLOBAL
B. REDRAW	B. REDRAW
C. ERASE	C. ERASE
DISPLAY OR EDIT SUBSTRUCTURES	DISPLAY OR EDIT SUBSTRUCTURES
D. ACTIVATE FOR DISPLAY	D. ACTIVATE FOR DISPLAY
	E. DISPLAY ATTRIBUTE
	F. DEACTIVATE
	G. RETAIN
	H. DRAW PICTURE
I. CREATE	I. CREATE
	J. DELETE FROM DATA BAS
	K. MERGE
L. EDIT SUBSTRUCTURES	L. EDIT SUBSTRUCTURES
M. RETURN	M. RETURN
a. First Appearance	b. Complete Menu

FIGURE 3.3. THE SUBSTRUCTURES MENU

The DEACTIVATE function is the reverse of the ACTIVATE FOR DISPLAY function (See Section 3.4.1). The options for selecting substructures to be deactivated are:

3.4.3.1 BY NAME.
3.4.3.2 BY ATTRIBUTE RANGE.
3.4.3.3 BY CROSSHAIRS.
3.4.3.4 ALL ACTIVE.

3.4.4 RETAIN

The RETAIN Function (under SUBSTRUCTURES) deactivates all active substructures except those selected with the crosshairs.

3.4.5 DRAW PICTURE

The DRAW PICTURE function (under SUBSTRUCTURES) displays all currently active items, including substructures, nodes, and attribute values. The screen is **not** automatically erased before the picture is drawn.

3.5 ELEMENTS

The ELEMENTS function specifies that elements in the data base are to be activated and displayed, and perhaps edited. (See Section 5.5 for editing elements). When ELEMENTS is selected, the system first displays the menu shown in Figure 3.4.a. The complete elements menu is shown in Figure 3.4.b.

3.5.1 DISPLAY ELEMENTS

This function allows the user to activate and display elements. (See Section 3.1 for more detail on activating). When elements are activated for display, they are automatically drawn. The DISPLAY ELEMENT function provides six options for identifying elements as discussed below.

A. GLOBAL	A. GLOBAL
B. REDRAW	B. REDRAW
C. ERASE	C. ERASE
DISPLAY OR EDIT ELEMENTS	DISPLAY OR EDIT ELEMENTS
D. DISPLAY ELEMENTS	D. DISPLAY ELEMENTS
	E. DISPLAY ATTRIBUTES
	F. DELETE FROM SCREEN
	G. RETAIN
H. CREATE	H. CREATE
I. DELETE FROM DATA BAS	I. DELETE FROM DATA BAS
J. EDIT ELEMENTS	J. EDIT ELEMENTS
K. RETURN	K. RETURN
a. First Appearance	b. Complete Menu

FIGURE 3.4. THE ELEMENTS MENU

After the elements have been identified (by one of the functions in Sections 3.5.1.1 thru 3.5.1.6), the screen is erased and the elements are displayed.

Once an element is activated it remains active until it is explicitly deactivated or deleted. The options for identifying elements are:

3.5.1.1 BY NUMBER
3.5.1.2 SEARCH ALL IN DATA BASE.
3.5.1.3 SEARCH ALL IN ACTIVE SUBSTRUCTURES
3.5.1.4 CONNECT NODES.
3.5.1.5 ALL IN ACTIVE SUBSTRUCTURES.
3.5.1.6 ALL IN DATA BASE.

3.5.2 DISPLAY ATTRIBUTE

The DISPLAY ATTRIBUTE function (under ELEMENTS) allows the user to select an attribute (such as element NUMBER, TEMPERATURE, or PRESSURE) and display the attribute value for one or more active elements. When DISPLAY ATTRIBUTES is selected, the list of all element attributes is displayed in the MENU AREA. The user is requested to identify the attribute whose value is to be displayed. (See Section 1.3.3 for instructions on selecting items from lists).

After the attribute is selected, the user identifies for which active elements the attribute value is to be displayed. This set of elements may be identified in several ways. Once the set is identified, the system displays the attribute value at the center of element. A two letter abbreviation for the attribute is also displayed (except for the attribute NUMBER). See Figure 3.5 for examples of displayed attributes.

If an attribute value is presently displayed, a special item appears in the list: DELETE ATTRIBUTE. When this list item is selected, the currently displayed attribute values are deleted from the display file. The screen is **not** automatically erased, but if REDRAW is selected, the picture is redrawn **without** the attribute values being displayed. If specified elements are deactivated, their displayed attribute values are also deactivated for display.

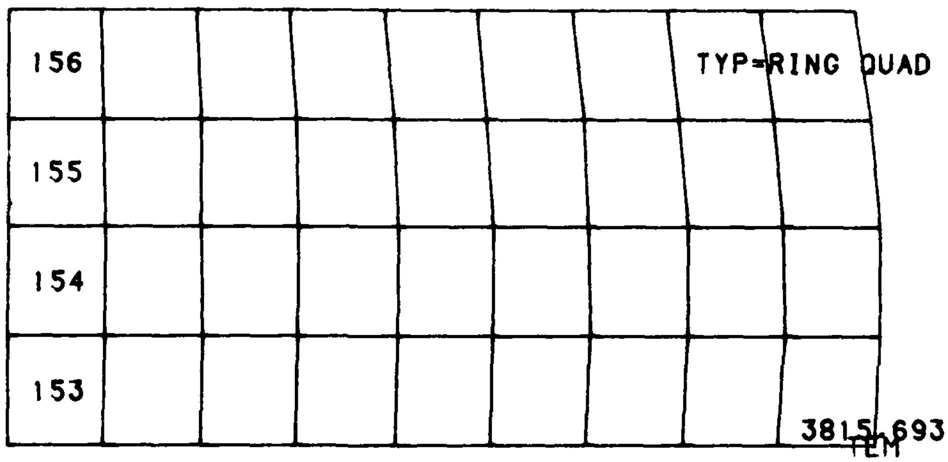


FIGURE 3.5. ATTRIBUTES DISPLAYED ON THE MODEL

If an attribute value is presently displayed, and a second attribute is selected, it will be displayed on top of the presently displayed attribute value. In general, this is not readable and the screen must be redrawn using REDRAW. Therefore, it is suggested that the user select ERASE before displaying a second attribute for an element which already has an attribute value being displayed. However, different attribute values can be selected and displayed for different elements without requiring the screen to be erased and the picture redrawn.

The methods for identifying the elements for which attributes values are to be displayed are:

3.5.2.1 BY CROSSHAIRS.
3.5.2.2 ALL ON SCREEN.

3.5.3 DELETE FROM SCREEN

The DELETE FROM SCREEN function (under ELEMENTS) identifies which elements are to be removed from the current set of active elements (that is, they are deactivated). It performs the reverse of the DISPLAY ELEMENTS function (See Section 3.5.1). When DELETE FROM SCREEN is used, the screen is not erased although the elements are removed from the display file. The user must select REDRAW after deactivating elements in order to make the picture current.

The user can select the elements to be deactivated with any of the following options:

3.5.3.1 BY NUMBER.
3.5.3.2 BY ATTRIBUTE RANGE.
3.5.3.3 BY CROSSHAIRS.
3.5.3.4 ALL ON SCREEN.

3.5.4 RETAIN

The RETAIN function (under ELEMENTS) specifies which elements in the current set of active elements are to remain active (that is, be

retained). The elements which are **not** retained are deactivated. When RETAIN is used, the screen is not automatically erased. A current picture with only active elements can be obtained by selecting REDRAW. The user identifies the elements to be retained by using the crosshairs.

3.6 NODES

The NODES function specifies that the user is displaying nodes. When NODES is selected, the system displays the menu shown in Figure 3.6.a. The complete nodes menu is shown in Figure 3.6.b.

3.6.1 DISPLAY NODES

DISPLAY NODES specifies the method by which nodes are identified for display. All nodes which are identifed become active nodes. (See Section 3.1 for more detail on activating.) The option for selecting nodes for display are:

3.6.1.1 BY ELEMENTS.
3.6.1.2 BY NUMBER
3.6.1.3 SEARCH ALL IN DATA BASE.
3.6.1.4 SEARCH ALL IN ACTIVE SUBSTRUCTURES.
3.6.1.5 ALL IN ACTIVE SUBSTRUCTURE.
3.6.1.6 ALL IN DATA BASE.

3.6.2 DISPLAY ATTRIBUTES

The DISPLAY ATTRIBUTE function (under NODES) specifies that an attribute value is to be displayed for one or more active nodes. The user can select nodes for displaying attributes with the following options.

3.6.2.1 BY CROSSHAIRS.
3.6.2.2 ALL ON SCREEN.

A. GLOBAL
B. REDRAW
C. ERASE

DISPLAY OR EDIT NODES

D. DISPLAY NODES

H. CREATE
I. DELETE FROM DATA BAS
J. EDIT NODES
K. RETURN

a. First Appearance

A. GLOBAL
B. REDRAW
C. ERASE

DISPLAY OR EDIT NODES

D. DISPLAY NODES
E. DISPLAY ATTRIBUTES
F. DELETE FROM SCREEN
G. RETAIN
H. CREATE
I. DELETE FROM DATA BAS
J. EDIT NODES
K. RETURN

b. Complete Menu

FIGURE 3.6. THE NODES MENU

3.6.3 DELETE FROM SCREEN

DELETE FROM SCREEN (under NODES) deactivates selected nodes and effectively erases them from the screen. The picture is not automatically redrawn; the user must select REDRAW in order to obtain a current picture. The options for identifying nodes to be deleted from the screen are:

3.6.3.1 BY NUMBER
3.6.3.2 BY ATTRIBUTE RANGE
3.6.3.3 BY CROSSHAIRS
3.6.3.4 ALL ON SCREEN

3.6.4 RETAIN

The RETAIN function (under NODES) retains user identified nodes and deactivates all other nodes from the list of currently active nodes. The user identifies nodes which are to remain active by using the crosshairs.

3.7 ERASE SCREEN

The ERASE SCREEN function automatically erases the screen and redraws the currently active menu. In addition, this function deactivates **all** active nodes. Active structures and substructures remain active after ERASE SCREEN is selected.

3.8 SUMMARY

STAGING has the capability to display both 2D and 3D models. There are numerous options for selecting what portion of a model to display. This can be accomplished by selecting the desired data base level (structure, substructure, element, or node) and then specifying which items to display. After a picture has been created, there are several capabilities for removing items. Figure 3.7 illustrates the STAGING Display menu tree. (Note certain functions included in Figure 3.7 are discussed in Chapter 5.)

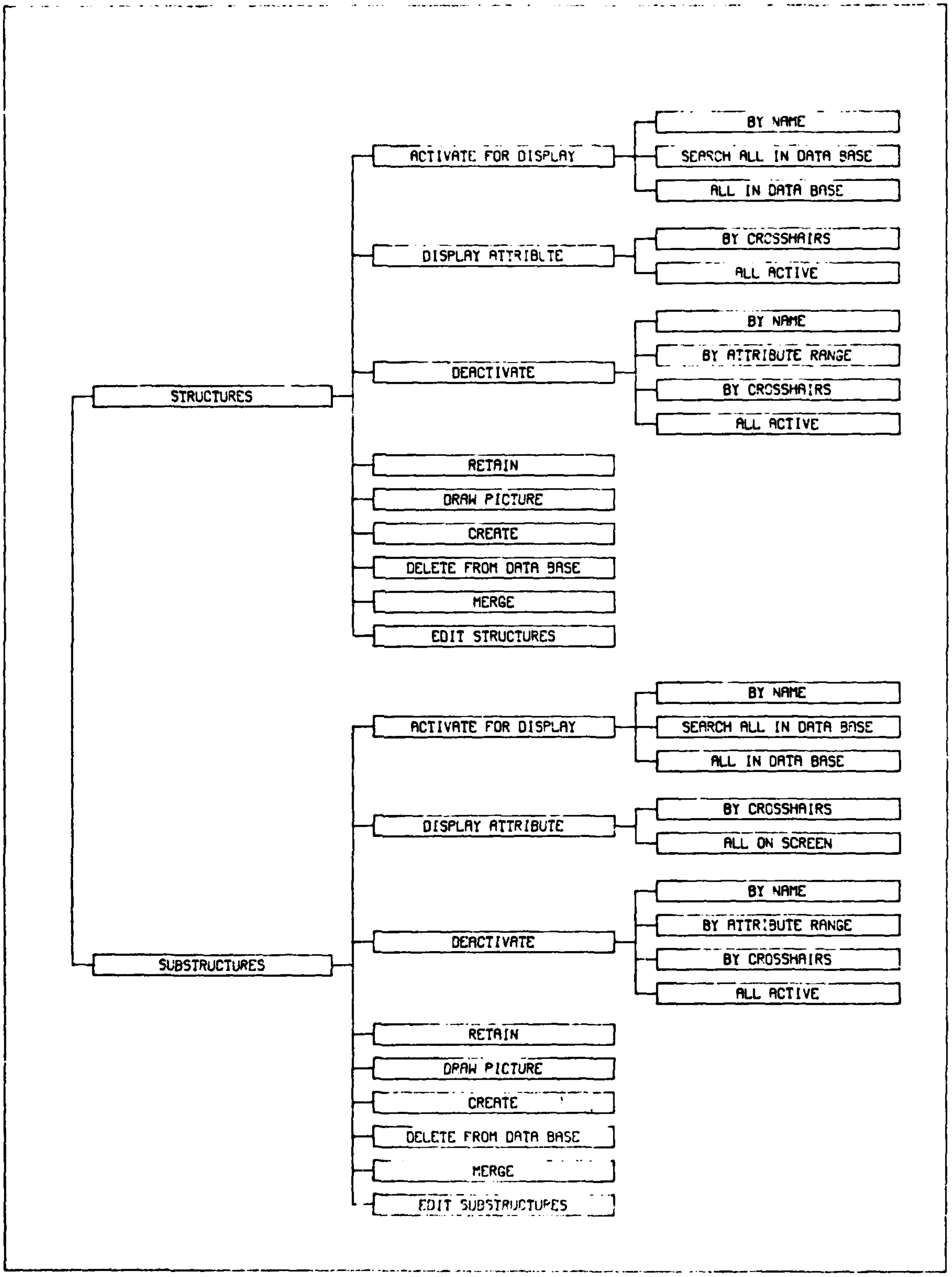


FIGURE 3.7.a. THE STAGING DISPLAY TREE

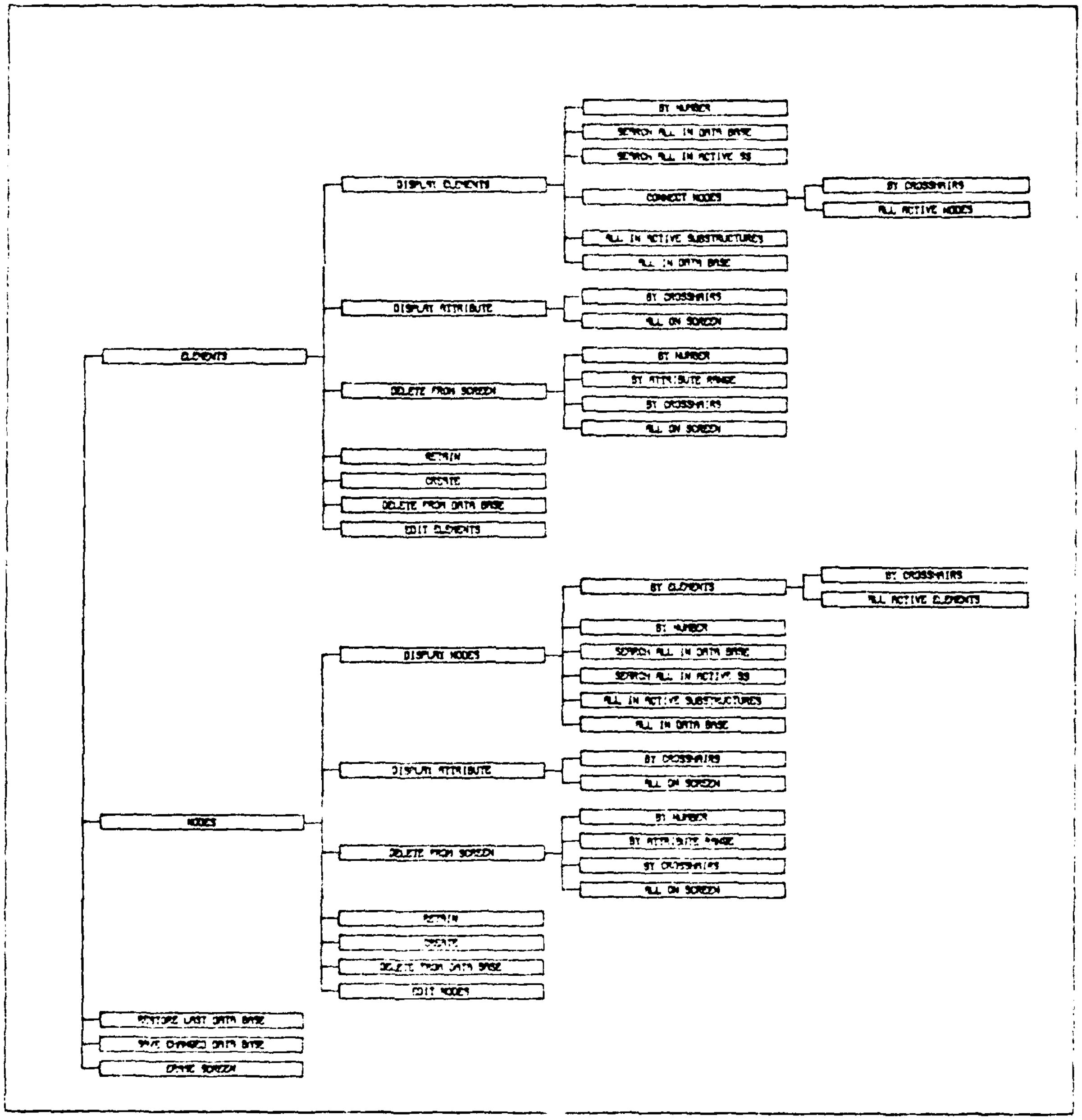


FIGURE 3.7.6. THE STAGING DISPLAY TREE

CHAPTER 4

MODIFYING THE PICTURE

This chapter discusses several functions which can be used to change the view of a model. This includes enlarging and compacting the image, recentering and filling the entire viewing area, and rotating the image in three space. There are also functions which shrink elements in order to allow the user to visually verify their connectivity. These functions are all located under the GLOBAL menu button. This button is always displayed at the top of the screen in the MENU AREA. (See Section 1.3)

The user should consult Chapter 3 to determine how to display an initial picture. Once the picture has been displayed, the functions in this chapter can be used to obtain the best viewing orientation for a given model.

4.1 RESTORE PICTURE

RESTORE PICTURE resets the scaling of the picture to its initial value of one (see Sections 4.2, 4.3, 4.4), restores the picture to its original center (see Section 4.5), unshrinks any reduced elements (see Section 4.6), and eliminates all multiple views (see Section 4.7). When RESTORE PICTURE is selected, the system automatically erases the screen and redraws the restored picture. RESTORE PICTURE does not affect changes made by rotating the model or specifying a perspective view.

4.2 COMPACT

COMPACT scales the view of the object in the display area by a factor of one half. That is, the resulting image is half the size of the preceeding view. The screen is automatically erased and the object is redrawn when COMPACT is selected. If the user has multiple images on the screen (see Section 4.7), **all** areas are redrawn after COMPACT, even though only one of the images may be compacted.

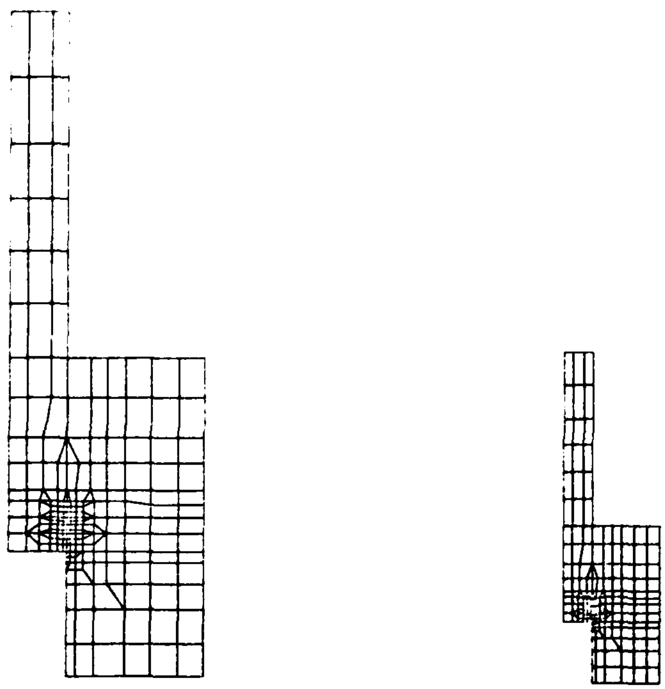
COMPACT is used to decrease the amount of object being viewed. In general, it is used after the original view of an object has been changed. COMPACT can be used to reverse the effect of ENLARGE (see Section 4.3) See Figure 4.1 for examples of COMPACT.

4.3 ENLARGE

ENLARGE scales the view of an object in the display area by a factor of two. That is, the resulting image is twice the size of the preceeding view. The screen is automatically erased and the object is redrawn when this button is selected. If the user has multiple images on the screen (See Section 4.2), all areas are redrawn after ENLARGE is selected.

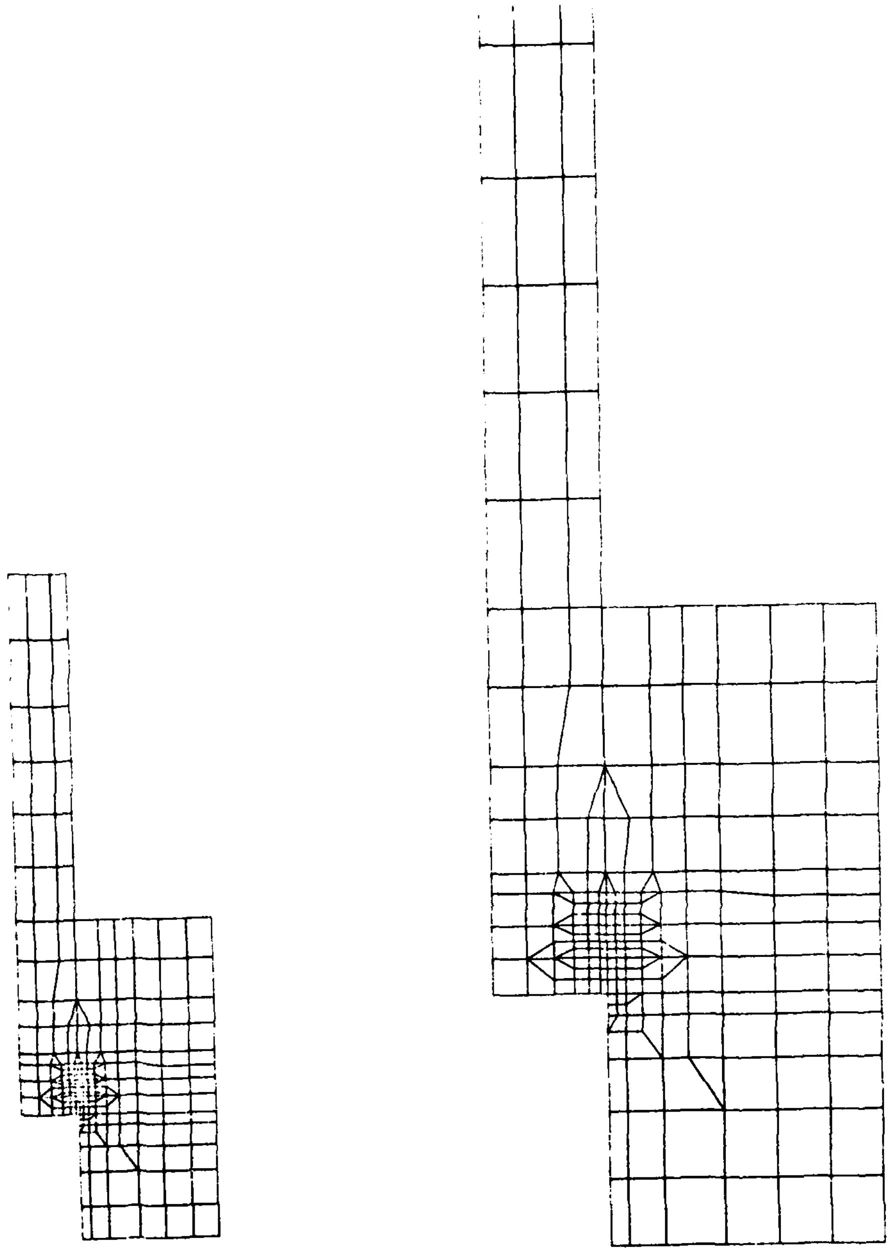
ENLARGE is used to inspect a model in closer detail. It can be used to reverse the effect of COMPACT. See Figure 4.2 for an example of ENLARGE.

ENLARGE should not be used to magnify a model with displayed attributes (see Section 4.8).



a. Normal b. Compact Once

FIGURE 4.1. EFFECT OF COMPACT BUTTON



a. Normal

b. Enlarge Once

FIGURE 4.2. EFFECT OF ENLARGE BUTTON

4.4 WINDOW

WINDOW allows the user to define a window around that portion of the object to be viewed. When window is selected, the crosshairs appear on the screen. The user selects a point in the display area which is any one of the four corners of the desired window. The system marks this point with the character "H". If the user picks a point outside of the display area, the "H" does not appear and the crosshairs reappear. The user must select the point again. The user then selects a point which is the opposite corner of the desired window. If the user picks a point outside of the display area, the crosshairs reappear. If the second corner is within the display area, the screen is automatically erased and the rescaled image is drawn.

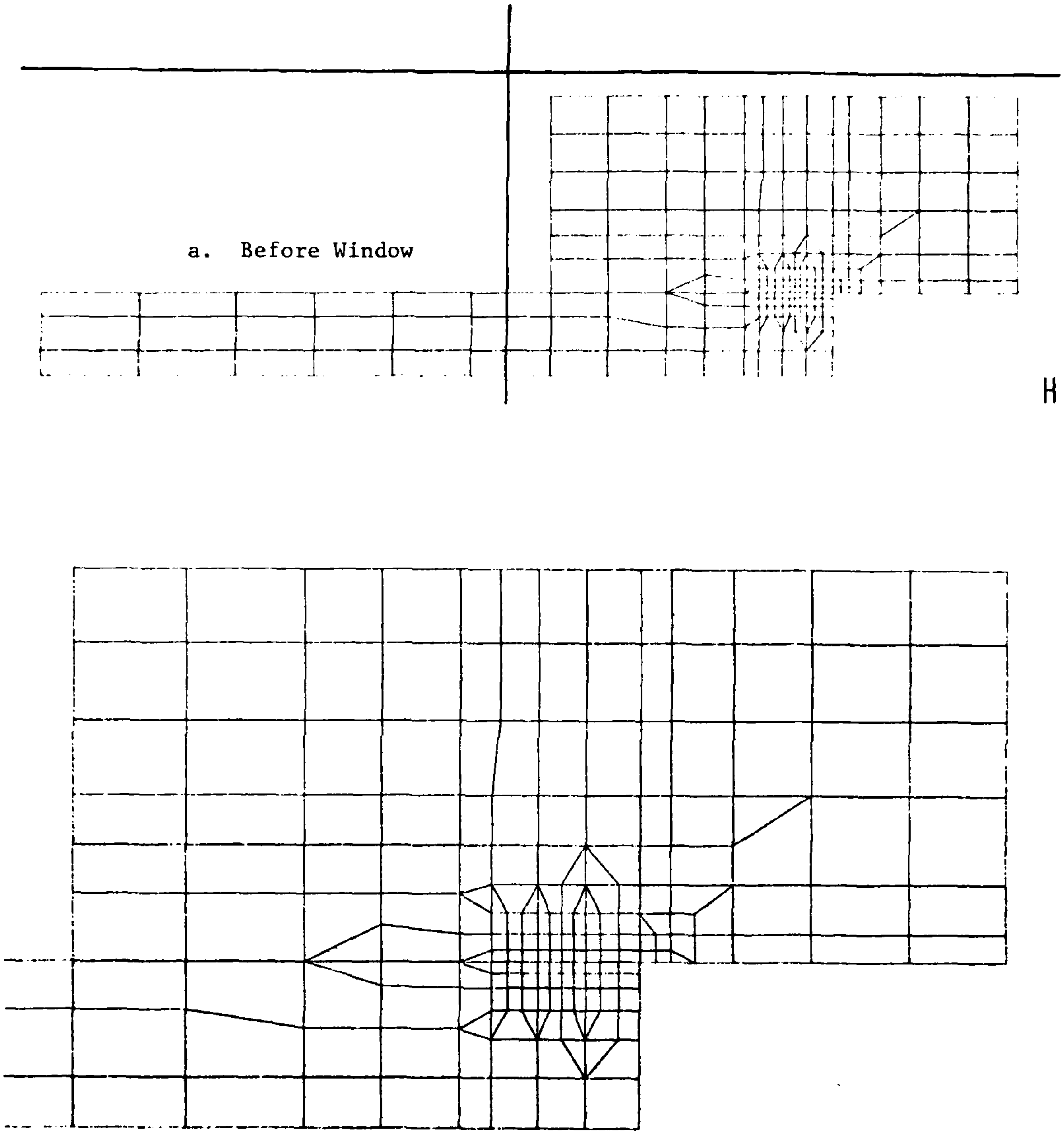
The system preserves the aspect ratio of the resulting image, and uses the largest dimension of the window to determine the dimensions of the resulting image. The center of the user selected window is the center of resulting image.

WINDOW is used to reduce the amount of detail on the screen. Enlarged views of an object allow the user to examine the object in closer detail. See Figure 4.3 for an example of WINDOW.

4.5 RECENTER

RECENTER allows the user to select a different center of view for an object being displayed. When RECENTER is selected, the crosshairs appear on the screen. The user selects a point in the display area which is to be the new center of view. If the user picks a point outside of the display area, the crosshairs reappear. If the user picks a valid point, the screen is automatically erased and the recentered picture is drawn. If the user has multiple images on the screen (See Section 4.7), all areas are redrawn after RECENTER.

RECENTER is used to include the desired view of an object in the display area. It can be used in conjunction with ENLARGE or WINDOW to translate across a large model. If the desired center of view is not in



a. Before Window

b. After Window

FIGURE 4.3. USING WINDOW

the display area, the user can pick the point in the display area which is closest to the desired center. This operation can be repeated until the desired center of view is in the display area. Alternatively, the user can select RESTORE PICTURE, select the desired center, and then ENLARGE. See Figure 4.4 for examples of RECENTER.

4.6 SHRINK

SHRINK reduces the size of the elements by approximately twenty percent. This function is used to inspect a model or part of a model to ensure that all coincident element boundaries have been defined.

When SHRINK is selected, the system presents a second menu from which the user selects ALL, SELECTED, or RETURN. (These buttons are described below.) Figure 4.5 shows the effect of the SHRINK function. **The screen is not automatically erased.** Therefore, the shrunken elements are superimposed on the unshrunken elements. The user can eliminate the unshrunken elements by selecting ERASE before selecting ALL, or by selecting REDRAW after selecting ALL.

4.6.1 ALL means that all active elements are to be reduced. If ALL is selected, the system draws reduced elements for all elements currently being displayed.

4.6.2 SELECTED means that only those elements identified with the crosshairs are to be reduced. After picking SELECTED, the crosshairs reappear. The user identifies all elements to be reduced and then selects RETURN. The selected elements will be reduced and drawn on top of the unshrunken elements. The system returns to the menu, allowing the user to select additional elements to be reduced. Thus, the user can iterate on shrinking selected elements until he is satisfied with the correctness of coincident boundaries.

If the user has activated a substructure and drawn the picture without explicitly activating elements (see Section 3.4.1), SELECTED refers to substructures instead of elements. That is, if the user identifies **any** element within the active substructure, **all** elements will be reduced.

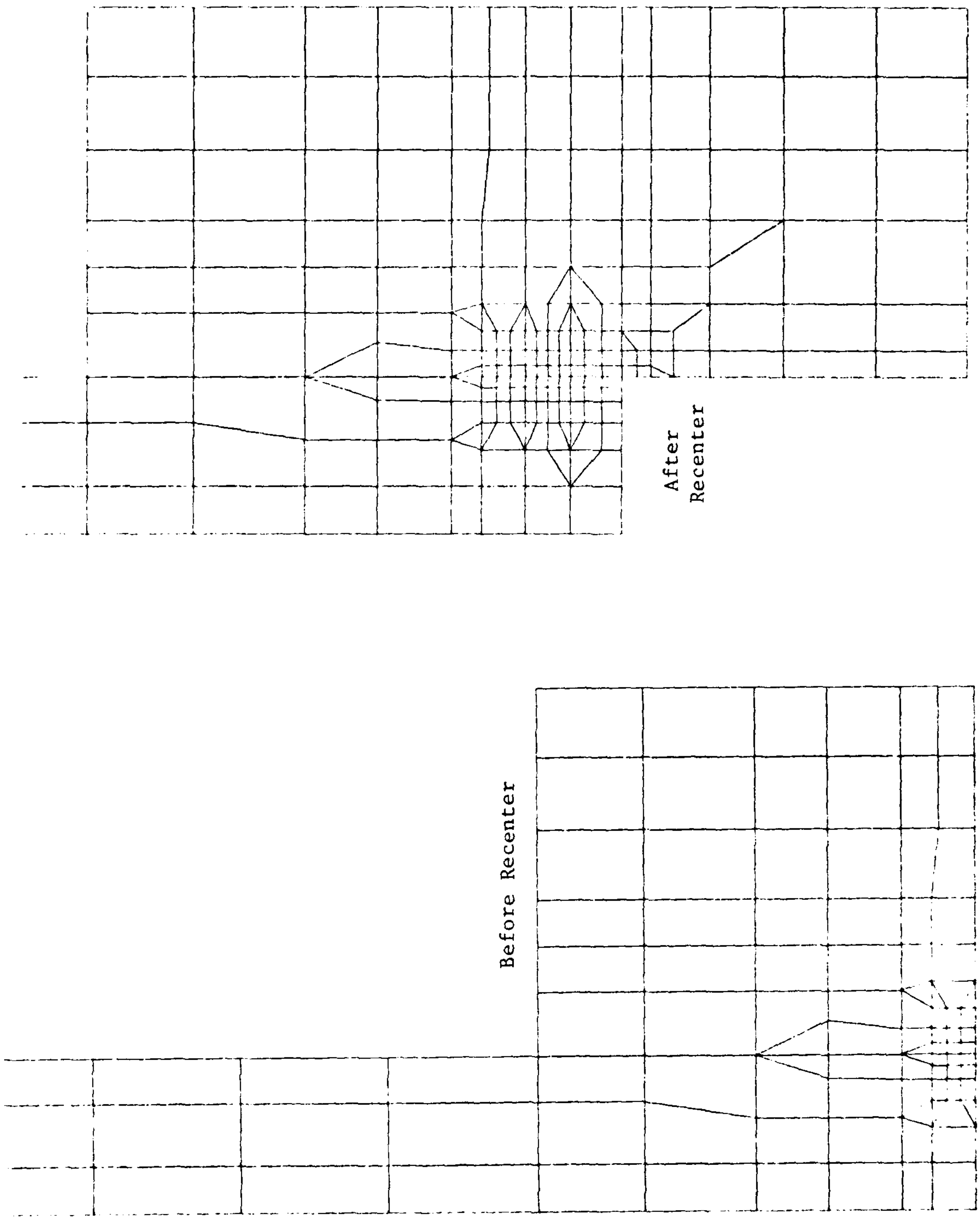
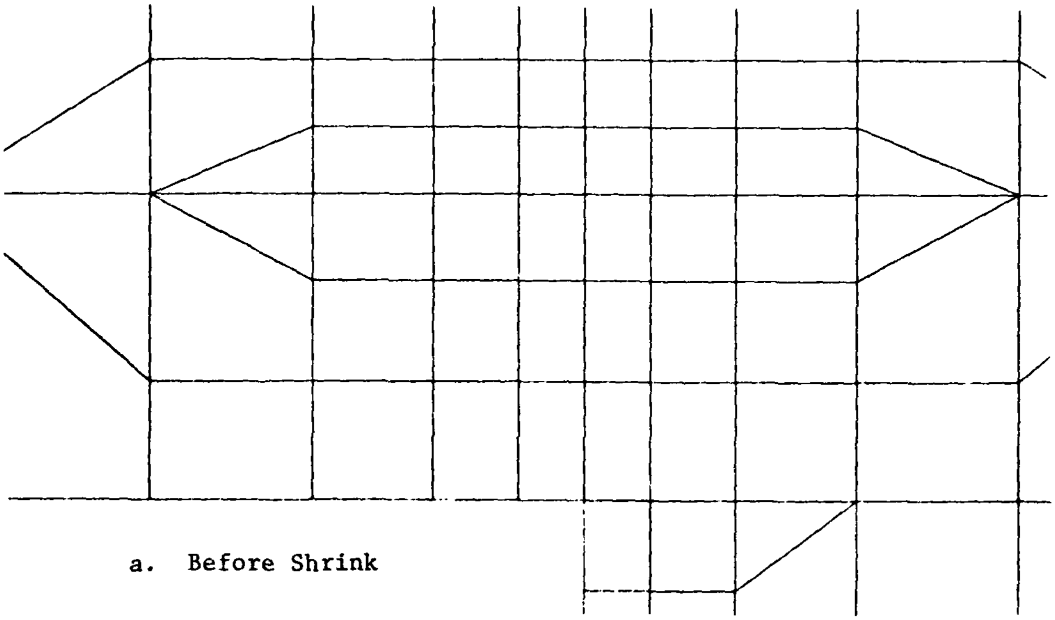
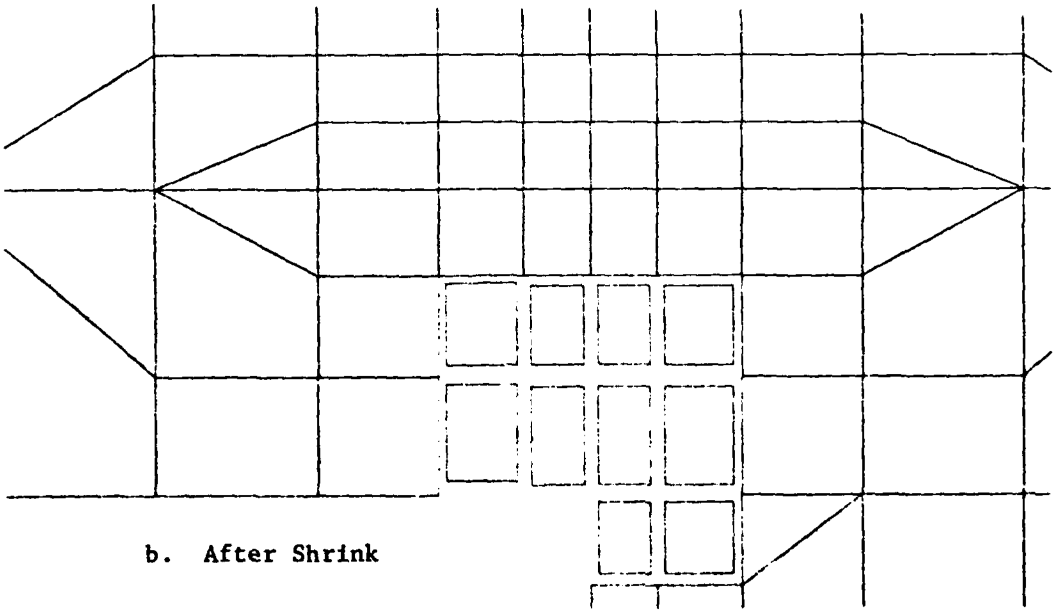


FIGURE 4.4. RECENTERING THE FIGURE



a. Before Shrink



b. After Shrink

FIGURE 4.5. SHRINKING ELEMENTS

AD-A089 381 BATTELLE COLUMBUS LABS OH F/G 13/13
STRUCTURAL ANALYSIS VIA GENERALIZED INTERACTIVE GRAPHICS - STAG--ETC(U)
SEP 79 L E HULBERT, C SCOFIELD F33615-76-C-3125
UNCLASSIFIED AFFDL-TR-79-3074-VOL-2 NL

2 OF 2
AD A089 381

END
DATE FILMED
10-80
DTIC

4.7 SPLIT SCREEN

The SPLIT SCREEN function defines up to four images on the screen at one time. A typical application is to display an entire model on the left side of the screen and use the right side to view the detail in an enlarged area. SPLIT SCREEN can be used to display four different views of the same model in engineering drawing fashion.

When SPLIT SCREEN is first selected, the screen is automatically erased and the image is reduced and redrawn in the primary display area (lower right screen quadrant).

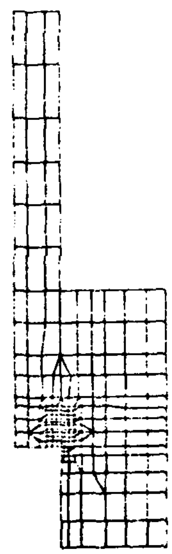
After the image is redrawn, there are buttons for restoring the view, and for associating or disassociating the new image.

4.7.1 RESTORE TO FULL erases the screen and redraws the image or images that were in the display area before SPLIT SCREEN was selected. RESTORE TO FULL does not reverse any viewing changes performed by other picture modifying functions.

RESTORE TO FULL is used after the user has viewed multiple images and desires to use the entire display area for viewing the object. RESTORE TO FULL can be used in case SPLIT SCREEN is picked by accident.

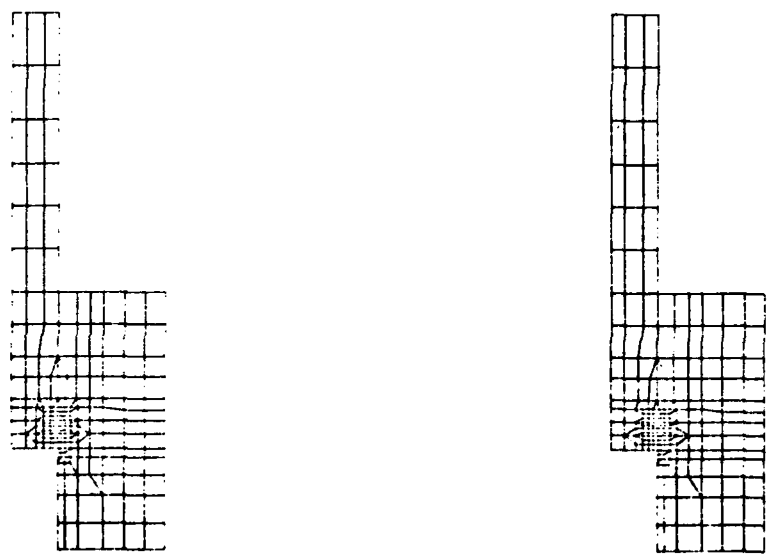
4.7.2 FREEZE LEFT SIDE first causes the newly created image to be added to the lower left screen quadrant. It is there fixed in its present orientation and scaling. After this button is selected, the new double image is drawn. Further manipulations of the view in the primary area do not effect the new image. Choosing SPLIT SCREEN and FREEZE LEFT SIDE a second or third time generates three or four images (see Figure 4.6). At each step, the frozen pictures are fixed, while the display in the primary area may be changed arbitrarily.

4.7.3 FREE LEFT SIDE allows the newly created image to be manipulated in conjunction with the view in the primary area. After this button is selected the new image is drawn. If any of the COMPACT, ENLARGE, WINDOW, or RECENTER functions are performed on the primary area,



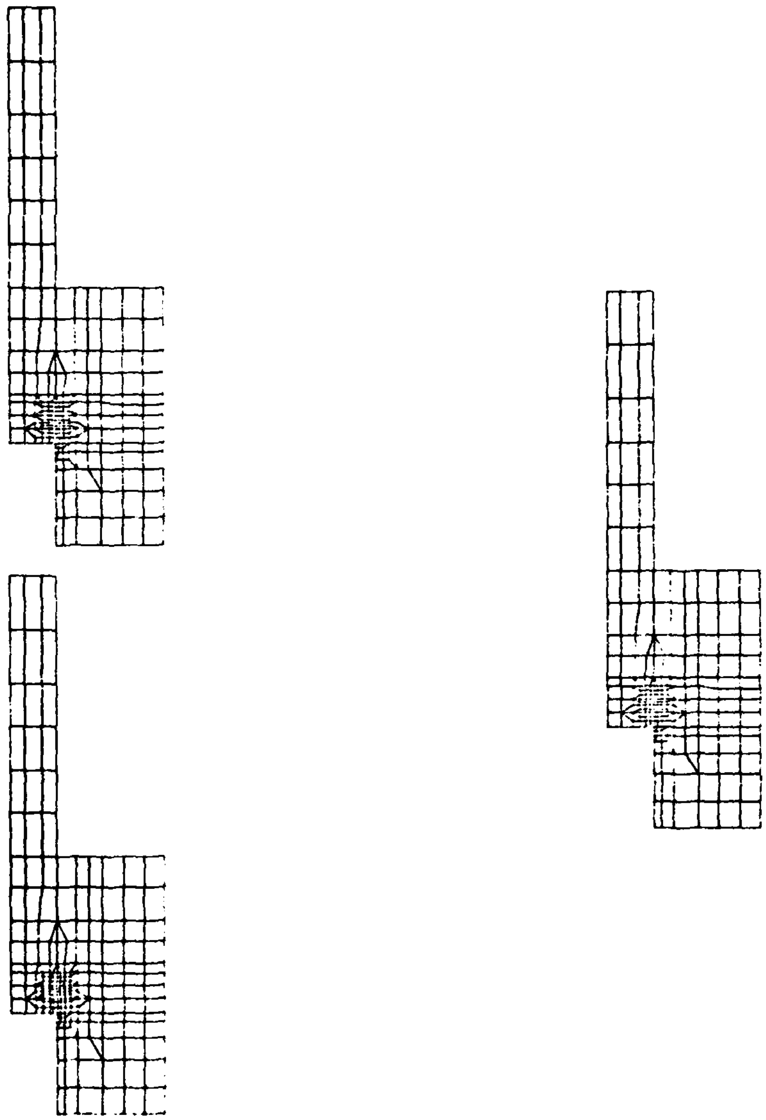
a. After first pick of SPLIT SCREEN

FIGURE 4.6. SPLIT SCREEN



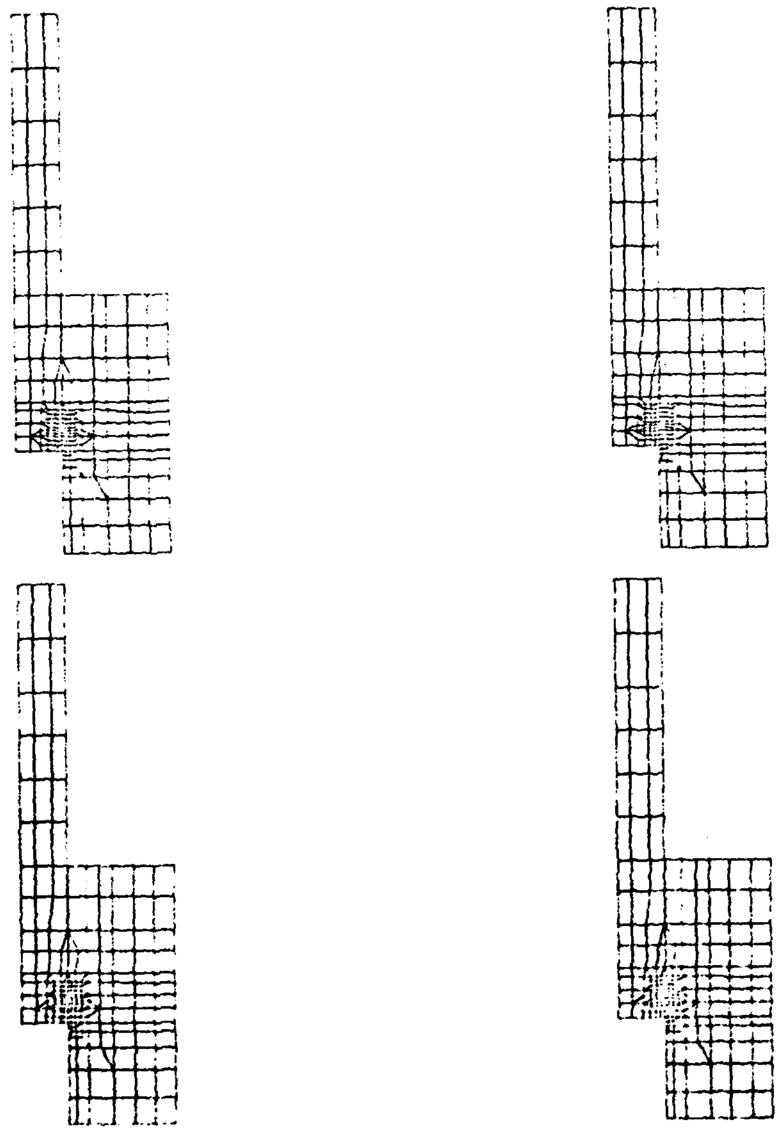
b. After One Pick of SPLIT SCREEN and Either FREEZE LEFT SIDE or FREE LEFT SIDE

FIGURE 4.6. SPLIT SCREEN
(Continued)



c. After Two Picks each of SPLIT SCREEN and FREEZE LEFT SIDE or FREE LEFT SIDE

FIGURE 4.6. SPLIT SCREEN
(Continued)



d. After Three Picks Each of SPLIT SCREEN and FREEZE LEFT SIDE or FREE LEFT SIDE

FIGURE 4.6. SPLIT SCREEN
(Continued)

these effects do not occur in a free image. If the primary view is ROTATED, the free image is also ROTATED. Multiple picks of SPLIT SCREEN and FREE LEFT SIDE generate multiple pictures (Figure 4.6).

4.7.4 RETURN returns to the previous menu without drawing any image in the SPLIT SCREEN. The area set aside for the new image remains unusable until the user selects RESTORE TO FULL or RESTORE PICTURE (see Section 4.1).

4.8 FILL SPACE

The FILL SPACE function increases the scale of the picture until it fits the entire screen. This function preserves the aspect ratio of the model in that all axes have the same units and the longest extend in the X, Y, or Z direction determines the length of the axes. Furthermore, the picture is centered in the DISPLAY AREA.

When FILL SPACE is selected the screen is automatically erased and the picture is redrawn. FILL SPACE must be used to enlarge pictures with displayed attributes. (See Figure 3.5).

The system initially sets the size of the space such that it will contain the entire model. If only a portion of the model is being viewed, FILL SPACE optimizes the use of the DISPLAY AREA.

4.9 CHANGE COORD SYSTEM

CHANGE COORD SYSTEM allows the coordinate system type to be changed. Normally, the system assumes that coordinates are three (or two) dimensional cartesian coordinates, or RECTANGULAR. CHANGE COORD SYSTEM allows the user to select one of the following.

4.9.1 RECTANGULAR.
4.9.2 POLAR.
4.9.3 CYLINDRICAL.
4.9.4 SPHERICAL.

4.10 AXES

The AXES function displays right-handed, three dimensional (x, y, and z) reference axes. When AXES is selected the axes are drawn from the point (0,0,0) and are labelled according to the model coordinate system.

If AXES is selected while the axes are displayed, they are turned off. The screen is not erased; the user can make the picture current by selecting REDRAW. Figure 4.7 illustrates the reference axes. The user can use the AXES function in order to determine the current orientation of the image after ROTATE or PERSPECTIVE has been performed.

4.11 ROTATE

The ROTATE function rotates three dimensional models in order to improve the user's viewing advantage. Two dimensional models can also be rotated if they are displayed as a three dimensional model with a zero Z coordinate (see Section 3.2).

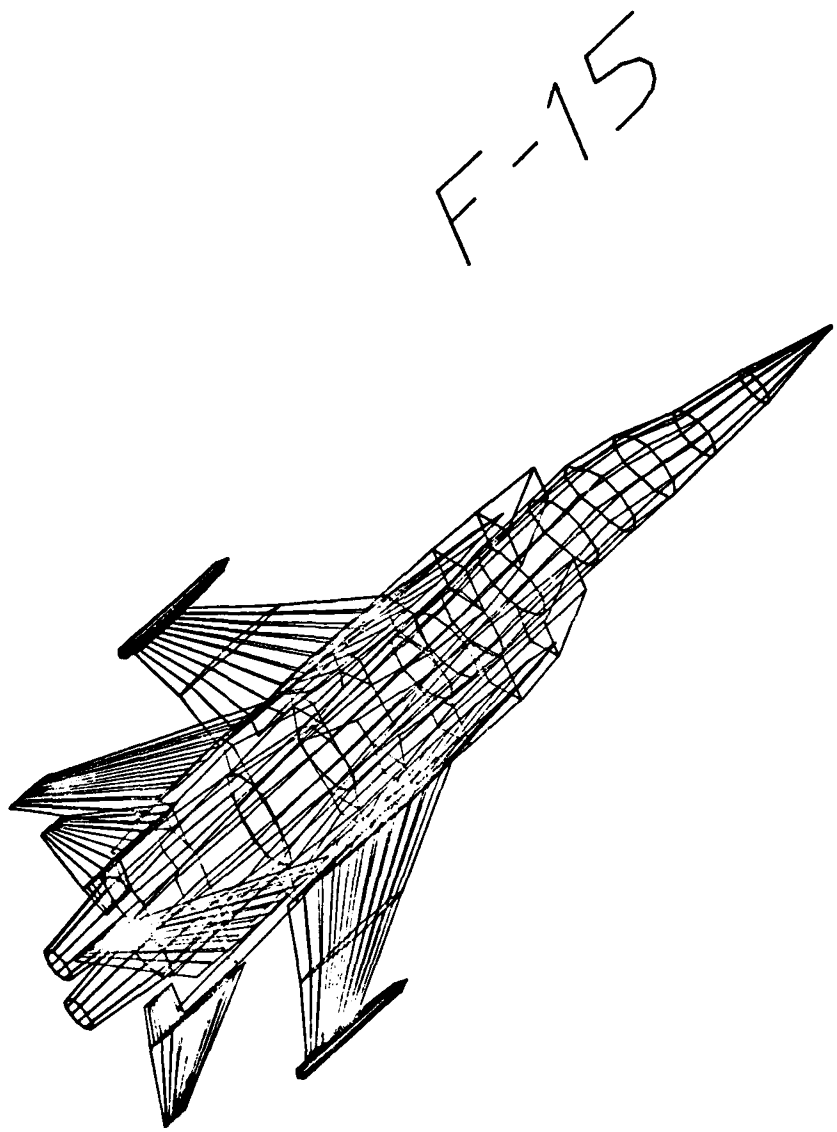
When ROTATE is initially selected, the system displays the data input template below:

A) X ANGLE=0
B) Y ANGLE=0
C) Z ANGLE=0

The angles of rotation entered by the user are specified in degrees. For repeated rotations, angles are added to any previous rotation angles.

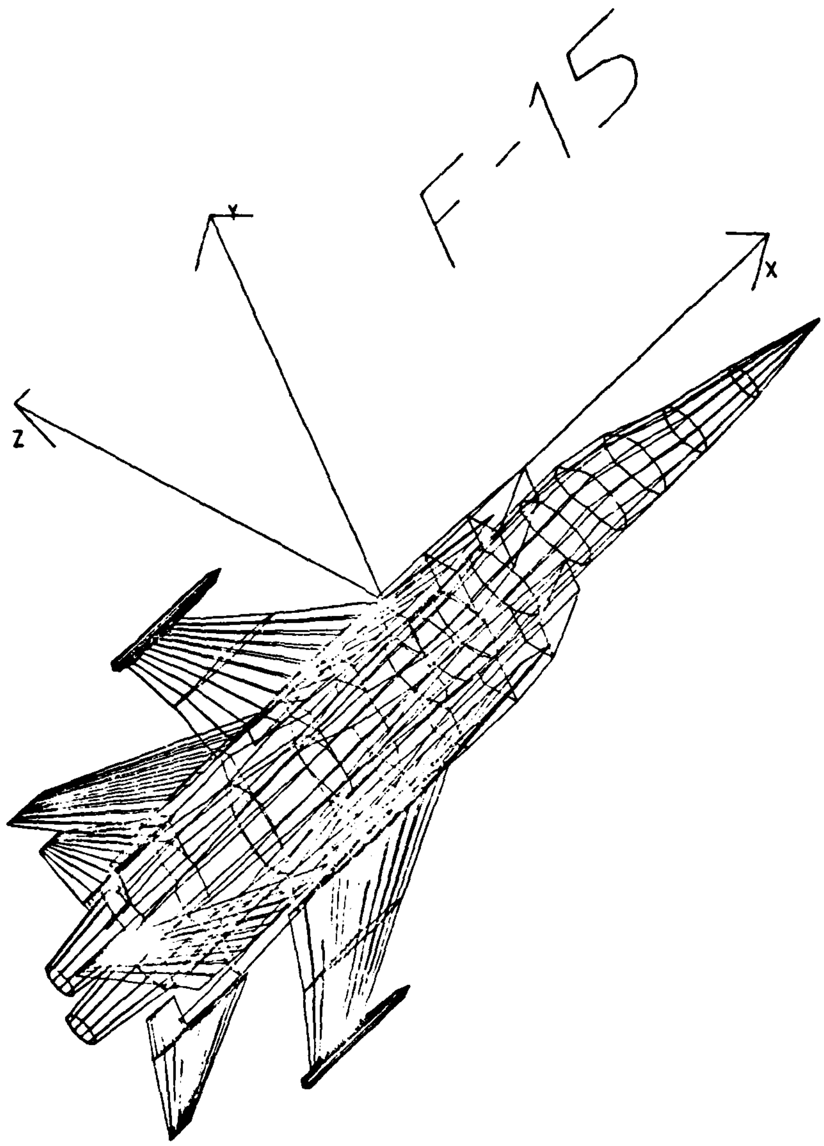
Therefore, if the user wishes to return to an unrotated view, he must enter the negatives of each of the three current rotation angles. After the angles are entered, the system automatically erases the screen and redraws the rotated image. If multiple images are on the screen (See Section 4.7), only the image in the primary area and images in freed associated areas are rotated.

The rotation performed by the system is **left-handed.** This means that if the user pretends to grasp a given axis with the **left**-hand such that his thumb points in the positive direction of the axis, then



a. No Axes

FIGURE 4.7. DISPLAYING AXES



b. Axes Shown

FIGURE 4.7. DISPLAYING AXES
(Continued)

positive type-in angles will rotate the image in the same direction as his fingers are curled. (This is clockwise if the user looks at his thumb.)

Initially, the model is displayed in a right-handed system with the positive X-axis to the "right", the positive Y-axis "up", and the positive Z-axis coming "out of the screen". (Figure 4.8) Thus, positive angles of rotation on the X-axis will tip the top of the model away from the user and bring the bottom forward. (Figure 4.9) Positive angles on the Y-axis will bring the right side of the model forward and push the left side back. (Figure 4.10) On the Z-axis, positive angles will turn the model clockwise; that is, the right side will tip down and the left side will tip up. (Figure 4.11) However, after the image has been rotated, the orientation of the axis is not to the right, up, and out of the screen. The user should use AXIS (See Section 4.10) in order to determine the current axes orientation.

4.12 PERSPECTIVE

The PERSPECTIVE function allows the user to define a perspective view of a three dimensional model. (This is a view in which the lines of the drawing appears to converge at a single point, the center of projection.) In addition, the user can specify **clipping planes** which exclude things "behind" the back plane and exclude things "in front of" the front plane.

When PERSPECTIVE is selected the following data input template is displayed:

A) LOOK AT X = 0
B) LOOK AT Y = 0
C) LOOK AT Z = 0
D) EYE POS X = 0
E) EYE POS Y = 0
F) EYE POS Z = + INFINITY
G) PROJ PLANE = 0
H) FRONT CUT = 0
I) BACK CUT = 0

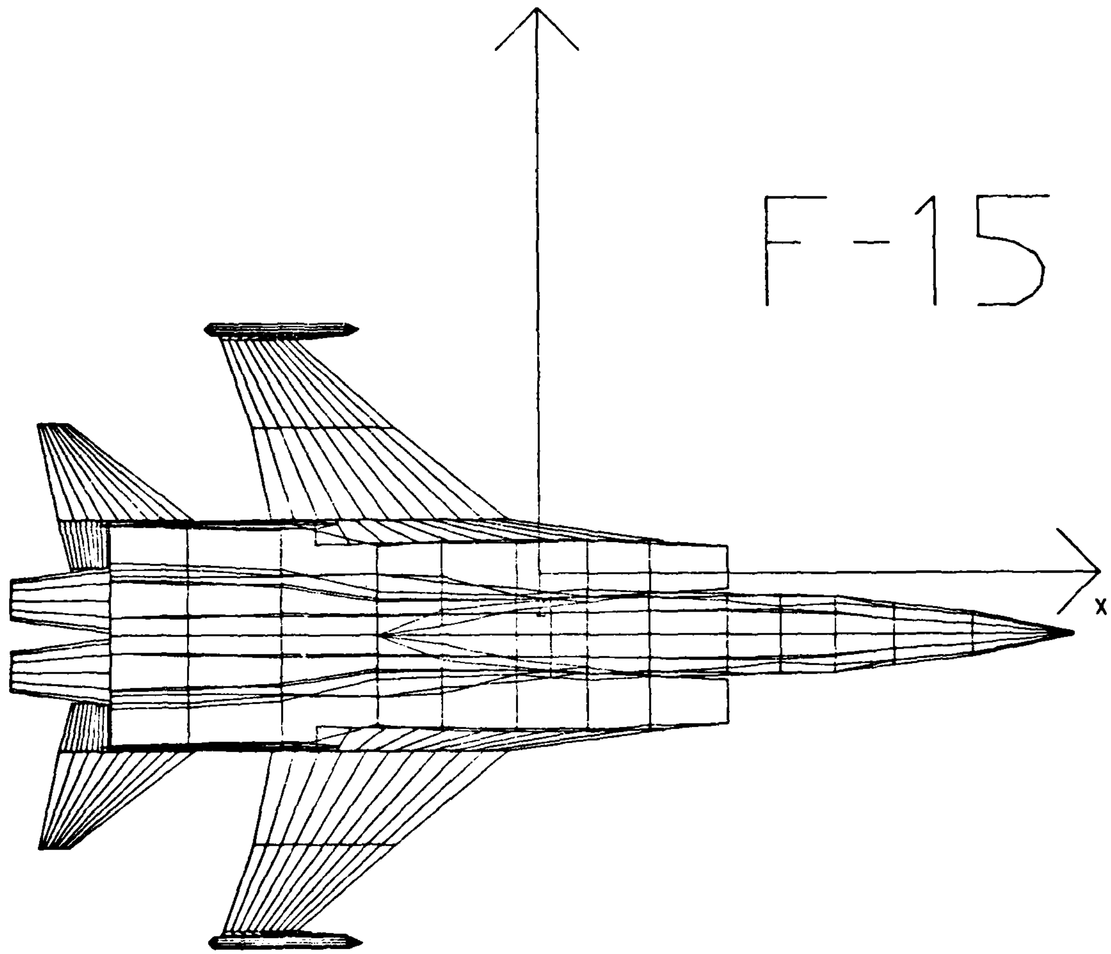


FIGURE 4.8. ORIENTATION OF THE MODEL AXES

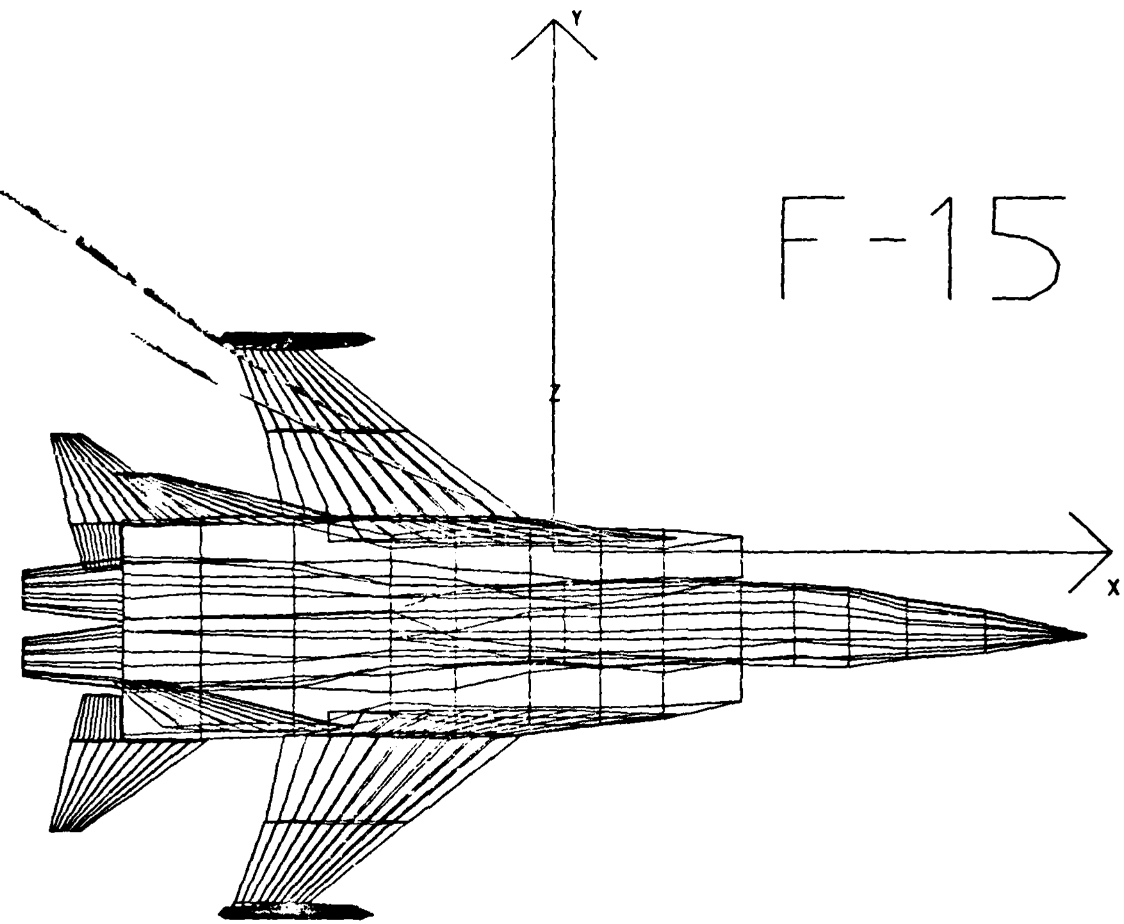


FIGURE 4.9. +20° ROTATION ABOUT THE X-AXIS

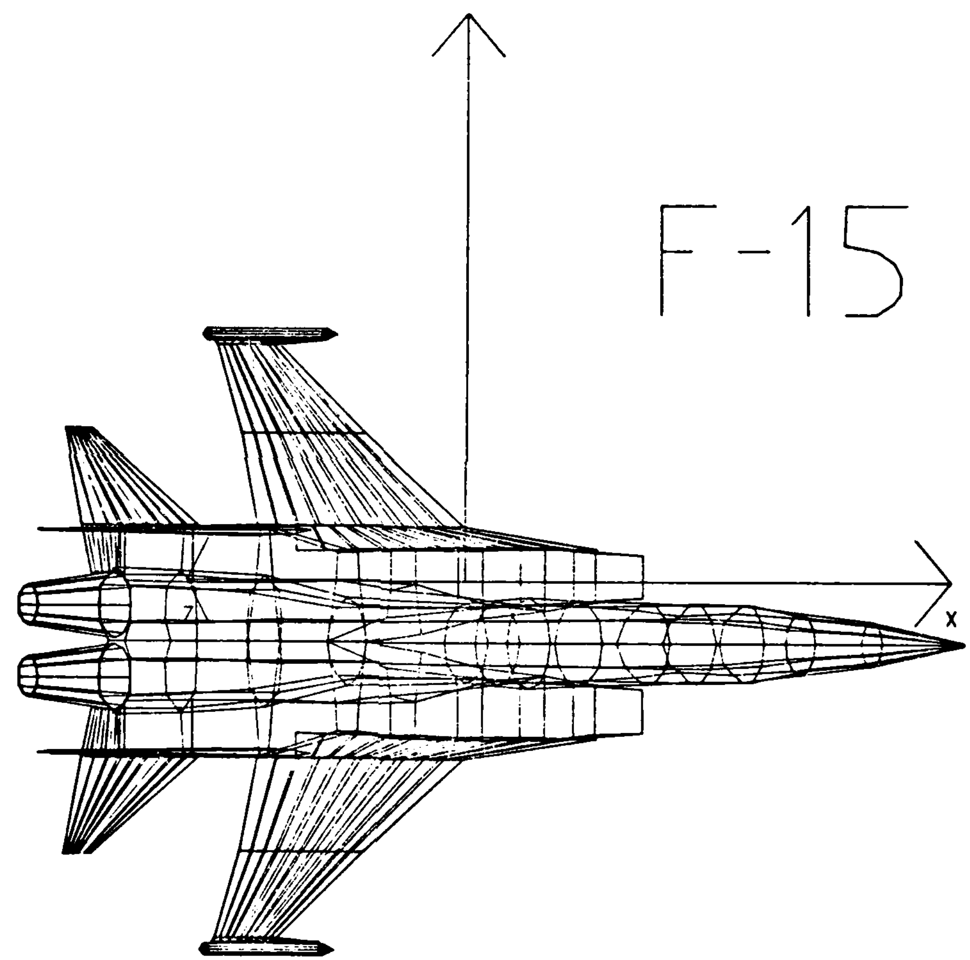


FIGURE 4.10. +30° ROTATION ABOUT THE Y-AXIS

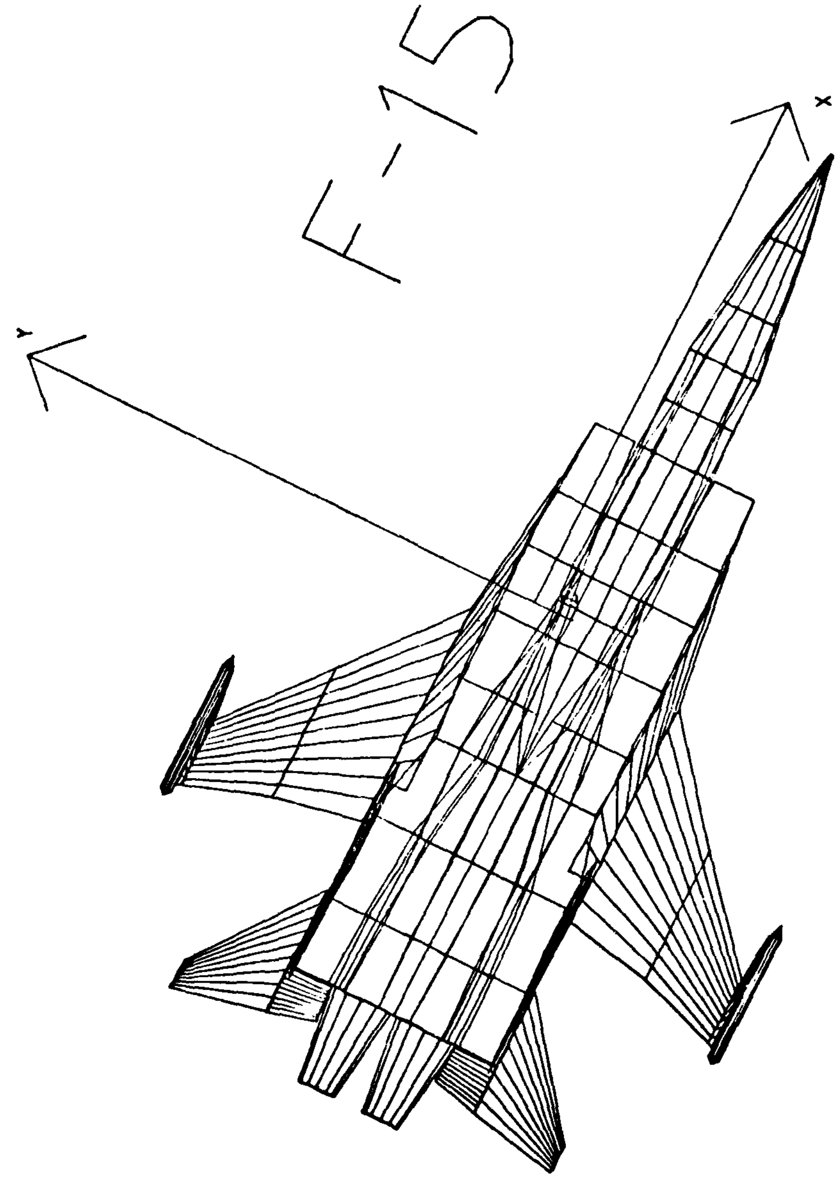


FIGURE 4.11. +30° ROTATION ABOUT THE Z-AXIS

These parameters specify a point to look at, an eye position, a projection plane, and front and back out planes. Table 4.1 describes these data items and their permissible values. Figure 4.12 illustrates how the parameters are used by the system.

PERSPECTIVE is a challenging and rewarding function to use. In general, the user should interactively vary the PERSPECTIVE parameters until he obtains the best and most realistic view of his model.

4.13 CHANGE VIEW PLANE

CHANGE VIEW PLANE specifies which principle plane to use for drawing three dimensional models in two dimensions. This allows the user to obtain plan views of three-dimensional models.

4.13.1 X Y PLANE uses the X-and Y-coordinates of the model and a zero Z coordinate. The view of the model is along the Z axis. This is the default for viewing a three dimensional model using 2D MODEL (see Section 3.2).

4.13.2 X Z PLANE uses the X and Z coordinates of the model and a zero Y coordinate. The view of the model is along the Y axis.

4.13.3 Y Z PLANE uses the Y-and Z-coordinates of the model and a zero X coordinate. The view of the model is along the X axis.

TABLE 4.1. PERSPECTIVE INPUT PARAMETERS

DATA ITEM	MEANING	PERMISSIBLE VALUES
LOOK AT X LOOK AT Y LOOK AT Z	X, Y, Z coordinates of the point to look at on the model.	An X, Y, Z on the model. If the character C [enter] is entered, the center of the model is calculated as the point looked at.
EYE POS X EYE POS Y EYE POS Z	X, Y, Z, coordinates of the eye	An X, Y, Z outside the volume occupied by the model. An entry of +[Infinity] or -[Infinity yields an orthogonal projection. C[enter] means the same as for point looked at.
PROJ PLANE	The projection plane. The closer the projection plane to the point looked at, the larger the picture.	A distance from the point looked at. If 0, a distance halfway between the point looked at and the eye is calculated.
FRONT CUT BACK CUT	Distances from the point looked at that control the cutoff planes. The front cut is the distance along +Z, the back cut along -Z.	Any positive value.

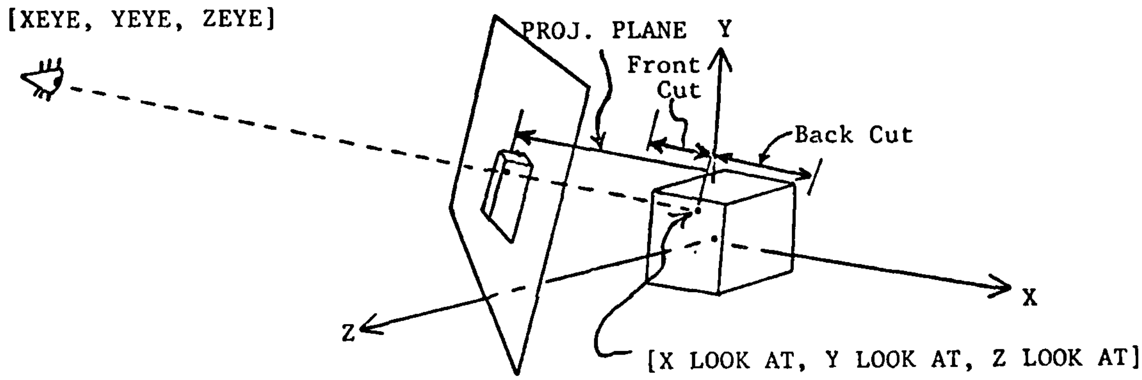


FIGURE 4.12. 3D VIEW CONTROL

4.14 Summary

STAGING contains very general capabilites for modifying a picture once it has been displayed. The user can change the scale of the picture, redefine the viewing window, shrink selected or all elements, rotate the image in three space, define a perspective projection, and redefine the type of coordinates. Figure 4.13 diagrams the STAGING menu tree for these picture modifying functions.

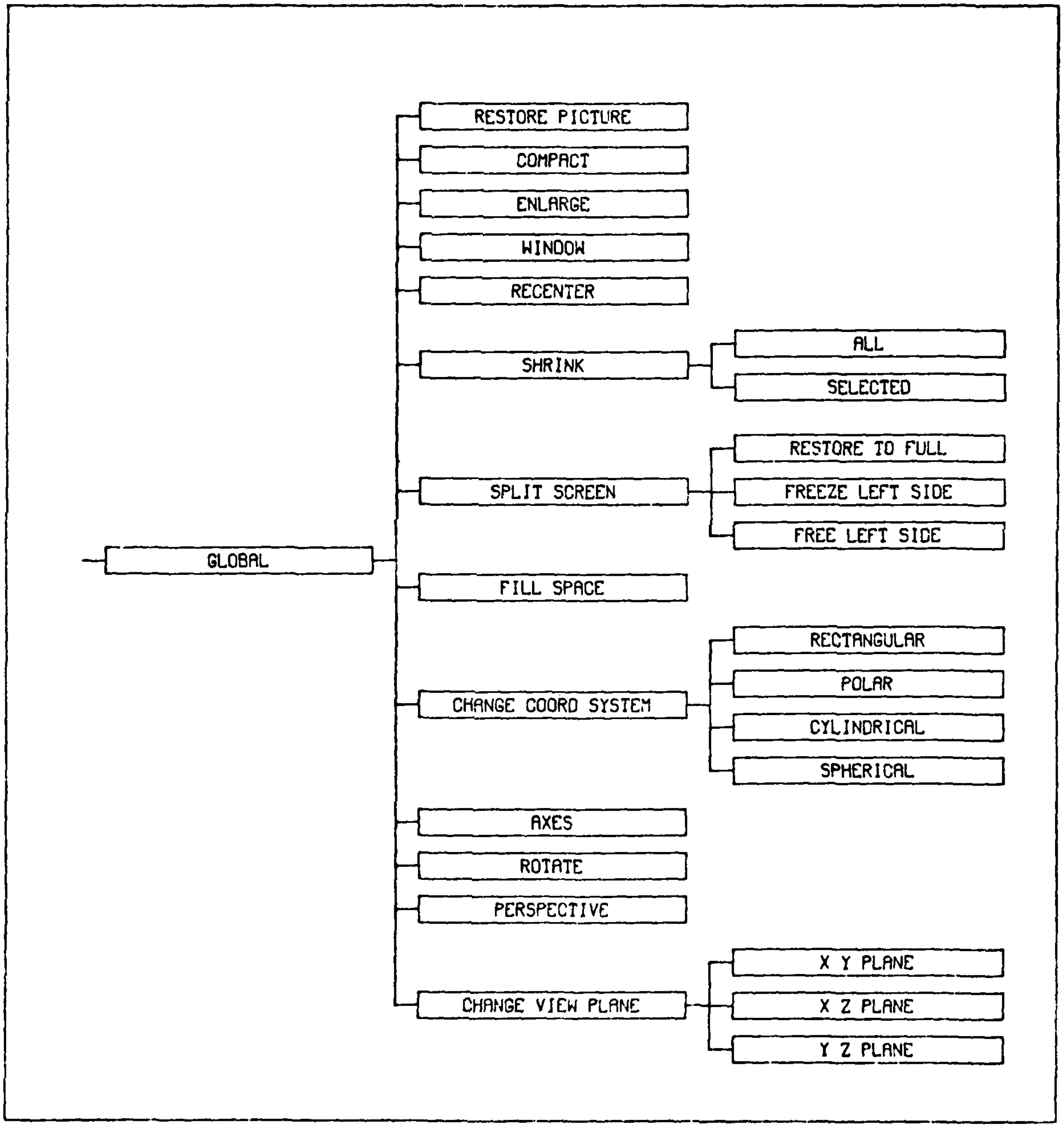


FIGURE 4.13. THE STAGING MODIFY PICTURE TREE

CHAPTER 5
EDITING MODEL DATA

This chapter presents the STAGING capabilities for correcting and modifying model data, including geometry, mesh, and attributes. These capabilities include functions which create new data base items, delete existing data base items, or change existing data base items. Additionally, there are functions for managing the changes to the data base in an orderly manner.

5.1 Managing Data Base Changes

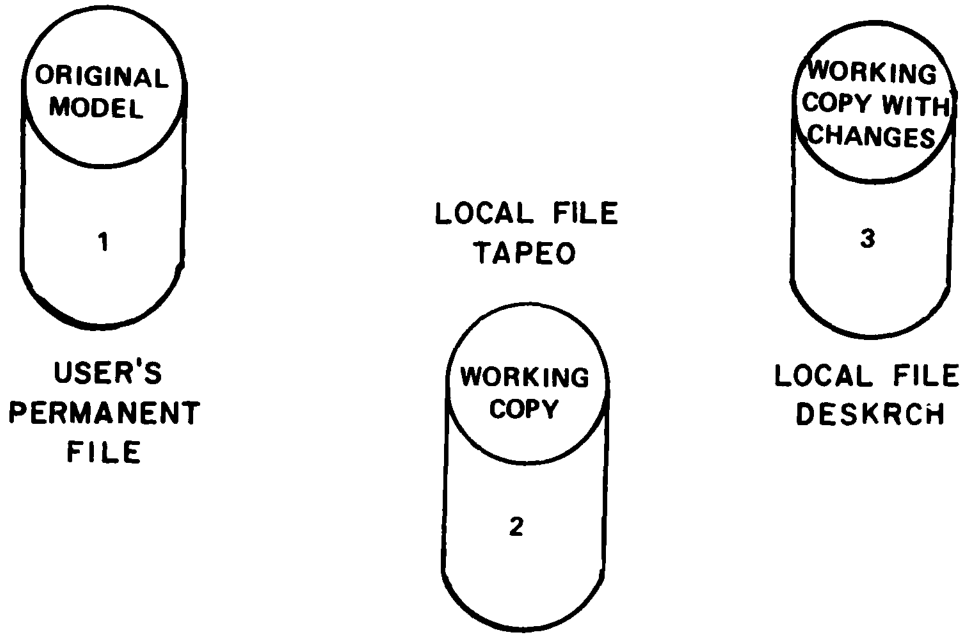
An understanding of the STAGING data base manipulations is necessary to allow the most efficient use of STAGING. In using STAGING, the user will be concerned with three copies of the STAGING data base. The first copy is on the users permanent file with his designated file name and ID. When the user signs into STAGING he is asked to type-in this PFN and ID (Section 1.5.3). Once STAGING verifies that the user has identified a valid data base, it makes a copy of the file to local file TAPEO. It also makes a second copy called DESKRCH. (See Figure 5.1.)

As the user interacts with STAGING to modify his data base, all changes are made to file DESKRCH. If the user makes a number of exploratory changes to his data base which he wishes to erase, he may pick RESTORE LAST DATA BASE. This command causes TAPEO to be copied onto DESKRCH so that the user returns to his original data base. In this operation, the current picture file is erased and the user must recreate a new display.

It is clear that the STAGING user will make changes to his data base that he wishes to save. As in any interactive environment, if the computing system crashes, the user's local files are lost and he must begin again with a permanent file. STAGING provides the user two ways to save his work.

In the first step, if the user has made a number of changes he wishes to save, he picks SAVE CHANGED DATA BASE. This causes file DESKRCH to be copied onto local file TAPEO. In this step the user makes a temporary snapshot of his data base to which he can return with RESTORE LAST DATA BASE.

To store the changed data base as a permanent file the user enters GLOBAL and picks either CATALOG AS CURRENT or CATALOG AS NEW. The first command erases the last cycle of the user's permanent file and replaces it with the current data base. The second command allows the user to catalog his data base either under a new file name or as a new cycle under the old file name. (See Section 8.2 and 8.3 for detailed discussions.)



'SAVE CHANGED DATA BASE' – REPLACES FILE 2 WITH FILE 3

'RESTORE LAST DATA BASE' – REPLACES FILE 3 WITH FILE 2

'SAVE DATA BASE' – ALLOWS FILE 3 TO BE CATALOGUED WITH A NEW FILE NAME OR AS A REPLACEMENT FOR FILE 1

FIGURE 5.1. DATA BASE FILES IN STAGING

It is strongly recommended that the STAGING user develop the habit of periodically saving intermediate results of his STAGING sessions. In so doing, he may save himself significant amounts of time in recreating work he has already done once.

5.2 STRUCTURES

Selecting the STRUCTURES button indicates that the user is creating, deleting or changing an existing structure.

5.2.1 CREATE

The CREATE function allows the user to define a new structure in terms of existing substructures and add it to the current data base. When CREATE is selected, the system displays a data input template and asks the user to enter a name (of up to 40 characters) for the new structure. The system verifies that the name is unique and does not presently exist. After establishing the name, the user can assign attribute values and add substructures to the structure.

5.2.1.1 ATTRIBUTES assigns attribute values to the newly created structure. Allowable attribute types are presented in template form and the user types in new attribute values.

5.2.1.2 ADD SUBSTRUCTURES identifies the substructures which define the newly created structure. The user identifies the substructures either BY CROSSHAIRS (if displayed) or BY NAME.

5.2.2 DELETE FROM DATA BASE

This function removes specified structures from the data base. Furthermore, all references to a deleted structure are also removed. The user identifies structures to be deleted either BY CROSSHAIRS (if they are displayed) or BY NAME.

Structures are deleted when the entire substructure linkage is no longer needed. Deleting a structure has no real impact on the visual (or geometric) characteristics of the model.

5.2.3 MERGE

The merge function creates a new, single structure from two or more existing structures. The structures which are merged are **not** deleted. The user identifies structures to be merged either BY CROSSHAIRS (if displayed) or BY NAME.

5.2.4 EDIT STRUCTURE

The EDIT STRUCTURE function has facilities for changing existing structures. Before editing is performed, the user must activate the desired structures for editing. After activating a structure, the user can change any attribute value and/or change the substructures linkage by adding or deleting substructures.

5.2.4.1 ACTIVATE FOR EDITING identifies the structures to be edited. (The process of activating for display is discussed in Section 3.1). The process is the same for editing items as it is for displaying items. The options available for identifying structures include:

1. BY CROSSHAIRS
2. BY NAME
3. ALL ACTIVE SUBSTRUCTURE
4. ALL IN DATA BASE.

5.2.4.2 ATTRIBUTES (SINGLE) allows the user to assign any or all attribute values to a **single** structure. This function is only available when a single structure is active; it is mutually exclusive with ATTRIBUTES (GROUP) (see Section 5.2.4.3).

When ATTRIBUTES (SINGLE) is selected, a data input template is displayed and the user enters new attribute values by type-in.

5.2.4.3 ATTRIBUTES (GROUP) allows the user to assign any or all attribute values to a group of active structures. The **same** attribute value is assigned to all of the structures. When this button is displayed, ATTRIBUTE(SINGLE) is hidden (see Section 5.2.4.2).

5.2.4.4 ADD SUBSTRUCTURES allows the user to identify the substructures that he wishes to add to the structure he has activated. Options available are BY CROSSHAIRS (if the substructure is displayed) or BY NAME.

Chosen substructures are fully linked to the activated structure with all data base pointers automatically established.

5.2.4.5 DELETE SUBSTRUCTURES FROM allows designated substructures to be disassociated from a given STRUCTURE. All links are deleted between the STRUCTURE and the SUBSTRUCTURES. However, the substructures are not removed from the data base. Substructures may be picked BY NAME.

5.3 SUBSTRUCTURES

The SUBSTRUCTURES level contains buttons that allow the user to create a new substructure or delete or change an existing substructure.

5.3.1 CREATE

The CREATE function (under SUBSTRUCTURES) allows the user to define a new substructure and store it in the data base. When CREATE is selected, the user enters a name for the new substructure and the system verifies that the name is unique. After naming the new structure, the user can initiate actions to establish the association of the substructure to other items in the data base.

5.3.1.1 ADD TO STRUCTURES specifies to which structures (possibly more than one) the new substructure is added. The user identifies the structures either BY CROSSHAIRS (if structures are displayed) or BY NAME.

5.3.1.2 ADD ELEMENTS TO selects the set of elements which define the newly created substructure. The user has seven ways to select the elements:

1. ALL ACTIVE ELEMENTS
2. BY CROSSHAIRS
3. ELEMENT NUMBER (type-in)
4. ATTRIBUTE VALUE (type-in)
5. RECTANGULAR (VOLUME), 3D mode
6. CYLINDRICAL (VOLUME), 3D mode
7. SPHERICAL (VOLUME), 3D mode
8. SELECT AREA , 2D mode
9. TYPE IN POINTS , 2D mode

These options may be used interactively. Each option may be triggered repeatedly or any sequence of options may be chosen to construct the group of elements the user wishes. Defining a substructure by element attribute value can be a very powerful means to segregate elements by material or by other features.

5.3.2 DELETE FROM DATA BASE

DELETE FROM DATA BASE (under SUBSTRUCTURES) removes selected substructures from the data base. All links to the owning STRUCTURES and to ELEMENTS are deleted. Note that since STAGING is working on a copy of the data base, deleted substructures may be restored by returning to the STRUCTURES level and picking RESTORE LAST DATA BASE. If the user picks SAVE CHANGED DATA BASE with the substructures deleted, he may no longer retrieve the substructures without reaquiring his original catalogued file.

See Section 5.1 for MANAGING THE DATA BASE copies.

5.3.3 MERGE

Merge substructures allows a new single substructure to be created by joining two or more substructures. The merged substructures

are not deleted. The user identifies the substructures to be merged BY CROSSHAIRS (if displayed) or BY NAME.

5.3.4 EDIT SUBSTRUCTURES

The EDIT SUBSTRUCTURES function has facilities for changing existing substructures. Before editing is performed the user must activate the desired substructure for editing. After activating a substructure, the user can change any attribute value and/or change the element linkage by adding or deleting elements.

5.3.4.1 ACTIVATE FOR EDITING identifies the substructures to be edited. (The process of activating is discussed in Section 3.1. The process is the same for editing items as it is for displaying items.) The options available for identifying substructures include:

1. BY CROSSHAIRS
2. BY NAME
3. ALL ACTIVE SUBSTRUCTURES
4. ALL IN DATA BASE

5.3.4.2 ATTRIBUTES (SINGLE) allows the user to assign any or all attribute values to a **single** substructure. This function is only available when a single substructure is active; it is mutually exclusive with ATTRIBUTES (GROUP) (see Section 5.3.4.3).

When ATTRIBUTES (SINGLE) is selected a data input template is displayed and the user is requested to enter new attribute values.

5.3.4.3 ATTRIBUTES (GROUP) allows the user to assign any or all attribute values to a group of active substructures. The **same** attribute value is assigned to all of the substructures. When this button is displayed, ATTRIBUTE (Single) is hidden (see Section 5.3.4.2).

5.3.4.4 Move SUBSTRUCTURES allows the user to translate the substructure to a new position by entering a new value of the substructure centroid. In 2D MODEL, the substructure centroid is computed and

displayed as a "C". The new centroidal position is chosen BY CROSSHAIRS. The entire substructure is then displayed. In the 3-D MODEL, STAGING computes the (X,Y,Z) coordinates of the SUBSTRUCTURE and presents these to the user in Template form. The user types in new centroidal coordinates. The substructure is then translated (without rotation) to bring its centroid to the typed in (X,Y,Z) position.

5.3.4.5 ADD TO STRUCTURES allows the substructure to be linked to any active structure. Options are BY ACTIVE STRUCTURE or BY NAME.

5.3.4.6 ADD ELEMENTS to allows the user to define the set of elements that he wishes to add to the substructure. The user has seven ways to add elements.

1. ALL ACTIVE ELEMENTS
2. BY CROSSHAIRS
3. ELEMENT NUMBER (TYPEIN)
4. ATTRIBUTE VALUE (TYPEIN)
5. RECTANGULAR (VOLUME)
6. CYLINDRICAL (VOLUME)
7. SPHERICAL (VOLUME)

(See Section 5.3.1.3)

5.3.4.7 DELETE FROM STRUCTURES allows the user to disconnect a substructure from a designated structure. All data base linkages with the structure are severed. However, the substructure is not deleted from the data base and all linkages to elements are retained. Options for identifying STRUCTURES are BY NAME and BY CROSSHAIRS (if the structure is displayed).

5.3.4.8 DELETE ELEMENTS FROM allows any set or sets of elements to be deleted from the activated substructure. The data base links between the elements and the substructure are broken. However, the elements are retained in the data base and will retain their links with other substructures. Options for selecting elements include:

1. BY LIGHTPEN
2. ALL ACTIVE ELEMENTS (in the substructure)
3. ELEMENT NUMBER (type-in)

5.4 ELEMENTS

Selecting the ELEMENTS button allows the user to CREATE, EDIT or DELETE individual elements.

5.4.1 CREATE

The CREATE function allows the user to define a new element. STAGING will show the user the highest element number (active or inactive) in the data base. The user is then prompted to define the new element number and then assign attributes and substructure nodal connections.

5.4.1.1 ATTRIBUTES displays a template of the element attribute list. Attributes are entered by type-in.

5.4.1.2 ADD TO SUBSTRUCTURES allows the user to define the substructures which he wants to attach the new element. Options are BY CROSSHAIRS (if displayed) or by NAME.

5.4.1.3 ADD NODES TO allows the user to define the set of nodes belonging to the newly created element. The NODES are entered in the element's nodal connection table in the order in which they are selected. NODES are selected BY CROSSHAIRS (if displayed) or BY NUMBER (type-in).

5.4.2 DELETE FROM DATA BASE

DELETE FROM DATA BASE will delete selected elements from current data base. Nodes connected to the deleted elements are not deleted. Options include BY CROSSHAIRS (if displayed), BY NUMBER (type-in), and ALL ACTIVE IN DATA BASE.

5.4.3 EDIT ELEMENTS

The edit elements level allows great flexibility in modifying the element or its attributes.

5.4.3.1 ACTIVATE FOR EDITING allows the user to select those elements he wishes to change. Selection options are BY CROSSHAIRS (if displayed), BY NUMBER (type-in), ALL ON SCREEN, ALL IN DATA BASE, and FOR ALL ACTIVE SUBST.

5.4.3.2 ATTRIBUTES (SINGLE) allows any defined attribute to be changed. STAGING draws a template of all attributes and their values. An attribute is changed by user type-in.

5.4.3.2 ATTRIBUTES (GROUP) allows attributes to be changed for all designated elements of the group. A template is presented of all attribute types for the element group. Type-in of a new attribute value resets this attribute for all elements of the group to the typed-in value.

5.4.3.3 MOVE ELEMENT allows an element to be translated by specifying a new position of the element centroid. In 2D MODEL the new centroid is positioned by crosshairs. In 3D MODEL, the new centroid position is given by type-in. The element is translated without distortion. However, each element which shares nodes with the moved element is stretched to confrom with the new postion of the moved element.

5.4.3.4 ADD TO SUBSTRUCTURE allows an element to be added to an existing substructure. Options are BY CROSSHAIRS (if displayed) and by NAME.

5.4.3.5 ADD NODES TO allows the user to add existing nodes to a given element. Options are BY LIGHTPEN (if nodes are displayed) or BY NUMBER (type-in). The element is redrawn with the new nodes as the last nodes in the nodal connection table for the element.

5.4.3.6 DELETE FROM SUBSTRUCTURE deletes activated elements from a designated substructure. Identification is made BY NAME of the substructure.

5.4.3.7 DELETE NODES FROM will allow nodes to be deleted from a singly activated element. Node numbers are displayed in menu form and are picked by crosshairs.

5.4.3.8 REPLACE NODES allows nodes in a single element activated for editing to be replaced. Current node numbers are displayed in a template and may be replaced one at a time or in a group by type-in.

5.5 NODES

Under the NODES option, the user may CREATE, EDIT, or DELETE nodes from his data base.

5.5.1 CREATE

To create a new node, the user first enters a unique node number not already in the data base. STAGING displays the highest defined node number for the user. Once the user defines the number, he should define the attributes and define the nodal position. In 2D MODEL the node is postioned by crosshairs. In 3D MODEL the nodal coordinates are entered by type-in.

5.5.1.1 ATTRIBUTES displays a template of all the nodal attribute names. The user defines these values by type-in. If attribute names are left undefined, later attempts to use them will cause program aborts.

5.5.1.2 ADD TO ELEMENTS allows the newly created node to be added to designated elements. Options are BY LIGHTPEN and BY TYPE-IN. The added node is the last node in the nodal connection table of the elements to which it is added.

5.5.2 DELETE FROM DATA BASE

Incorrect nodes may be deleted from the data base. Options are BY CROSSHAIRS (if displayed) or BY NUMBER (type-in). Once the nodes are deleted, all elements to which these nodes were attached are redrawn with only their remaining nodes.

5.5.3 EDIT NODES

Edit Nodes allows individual or groups of nodes to be changed.

5.5.3.1 ACTIVATE FOR EDITING allows selected nodes to be activated. Selection options are BY CROSSHAIRS (if displayed), BY TYPE-IN, ALL ON SCREEN, ALL IN DATA BASE, or FOR ALL ACTIVE SUBSTRUCTURES.

5.5.3.2 ATTRIBUTES (SINGLE) displays a template of all defined attributes for this node and their values. The user may enter by type-in a changed value for any of the attributes. This option is sometimes useful just to display the attribute values.

5.5.3.1 MOVE NODES (SINGLE) allows the user to move a node to a new position. In 2D MODEL, this is done by crosshairs on the screen. In 3D MODEL the user types in new values of the (X,Y,X) coordinates. All elements connected to the node are redrawn in their new configuration. The screen is not erased so REDRAW should be picked to see the current shape.

5.5.3.3 ADD TO ELEMENTS (SINGLE) allows the node to be added to the nodal connection table of an element. The node becomes the last node in the node table of the element and the element is redrawn with the added node. Options for choosing the element are BY CROSSHAIRS (if displayed) or BY NUMBER (type-in).

5.5.3.4 DELETE FROM ELEMENTS (SINGLE) allows the node to be deleted from a selected element. The element is redrawn automatically with the deleted nodes removed and with closure on the remaining nodes. Options include BY CROSSHAIR (if displayed) or BY NUMBER (type-in).

5.5.3.5 ATTRIBUTES (GROUP) displays a template of the attribute names defined for nodes. Typing in a value for one of the attributes will reset the value of this attribute for the entire group of nodes to the typed-in value.

5.6 RESTORE LAST DATA BASE

STAGING provides the user with the means to work with a copy of his data base so that he can investigate the effects of various changes in his model without destroying his original data base. If a change is made in error or the user simply is dissatisfied with the changes be has made, picking RESTORE LAST DATA BASE will recopy the user's original data base (stored as TAPEO) onto the STAGING working file. In this operation, the display is erased and the user can proceed to construct a new display with the original data base.

5.7 SAVE CHANGED DATA BASE

If the user has made a set of changes or corrections in his model that he wishes to save, he can pick SAVE CHANGED DATA BASE. This will copy the STAGING working file onto the file TAPEO. This operation will overwrite the data base that had been on TAPEO. Thus, if the user makes further changes in his model and then picks RESTORE LAST DATA BASE, he will obtain the data base that he had saved with his call to SAVE CHANGED DATA BASE. By proper use of these SAVE and RESTORE commands, the user has great flexibility to make changes in his data base without destroying what he has already generated. At the same time he can save what he has accomplished any time that he wishes.

Section 5.1 includes further discussion of the use of these two commands together with the GLOBAL catalog data base command in managing the user's data bases.

5.8 SUMMARY

STAGING provides the user virtually unlimited power to change the model stored in his data. Many of the editing operations may be carried out in more than one way. Thus the user may add a node to an element by editing an element and picking ADD NODES TO, or he may add the node to the element by editing the node and picking ADD TO ELEMENT.

Attributes may be changed either for single nodes or elements or by groups of nodes or elements. It is important to remember, however, that if an attribute value is typed in for a group of elements or nodes all values of that attribute are set to the typed-in value for the activated group of elements or nodes.

Figure 5.1 illustrates the STAGING EDIT trees for each of the STAGING data base levels.

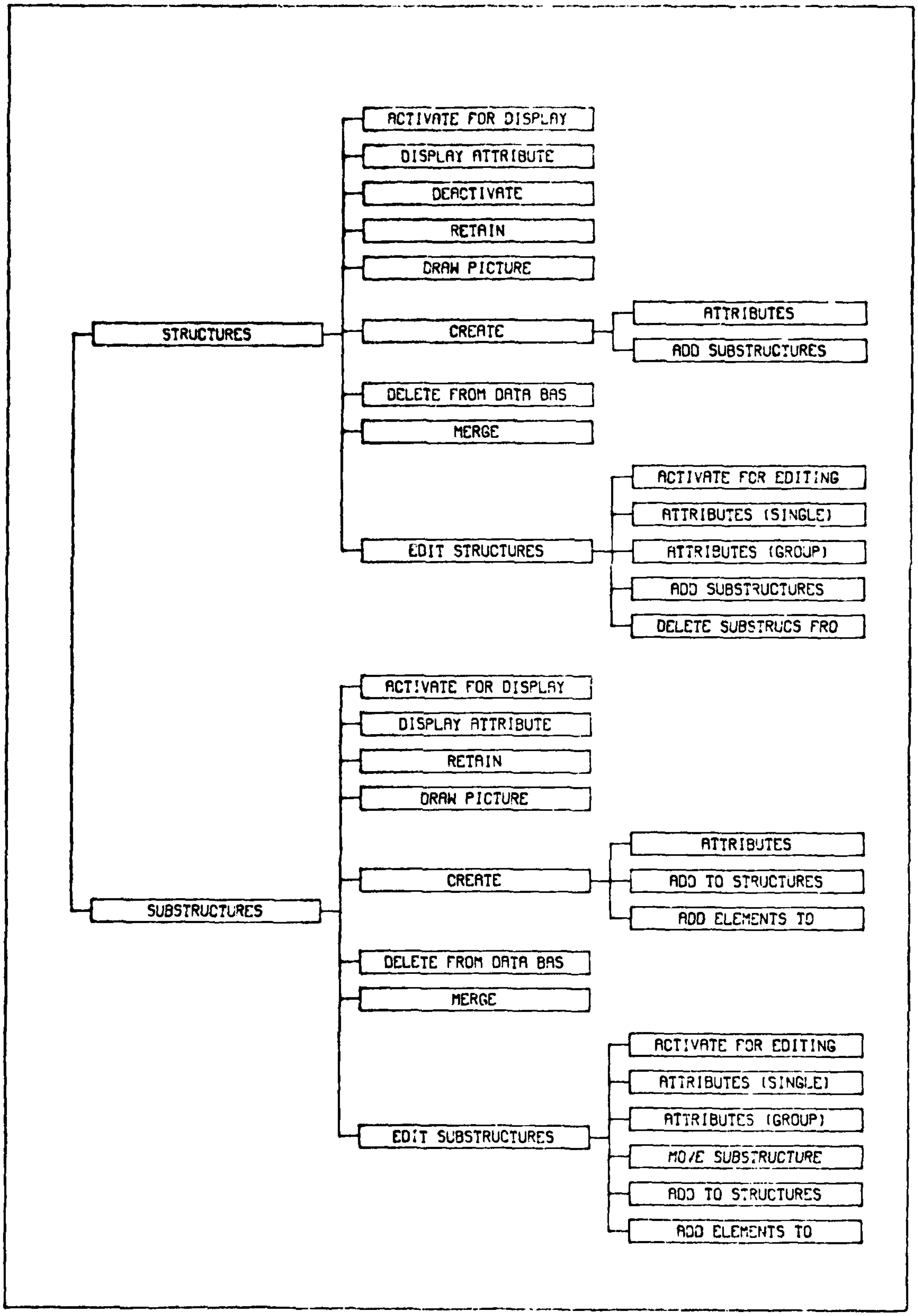


FIGURE 5.2.a. THE STAGING EDIT TREE

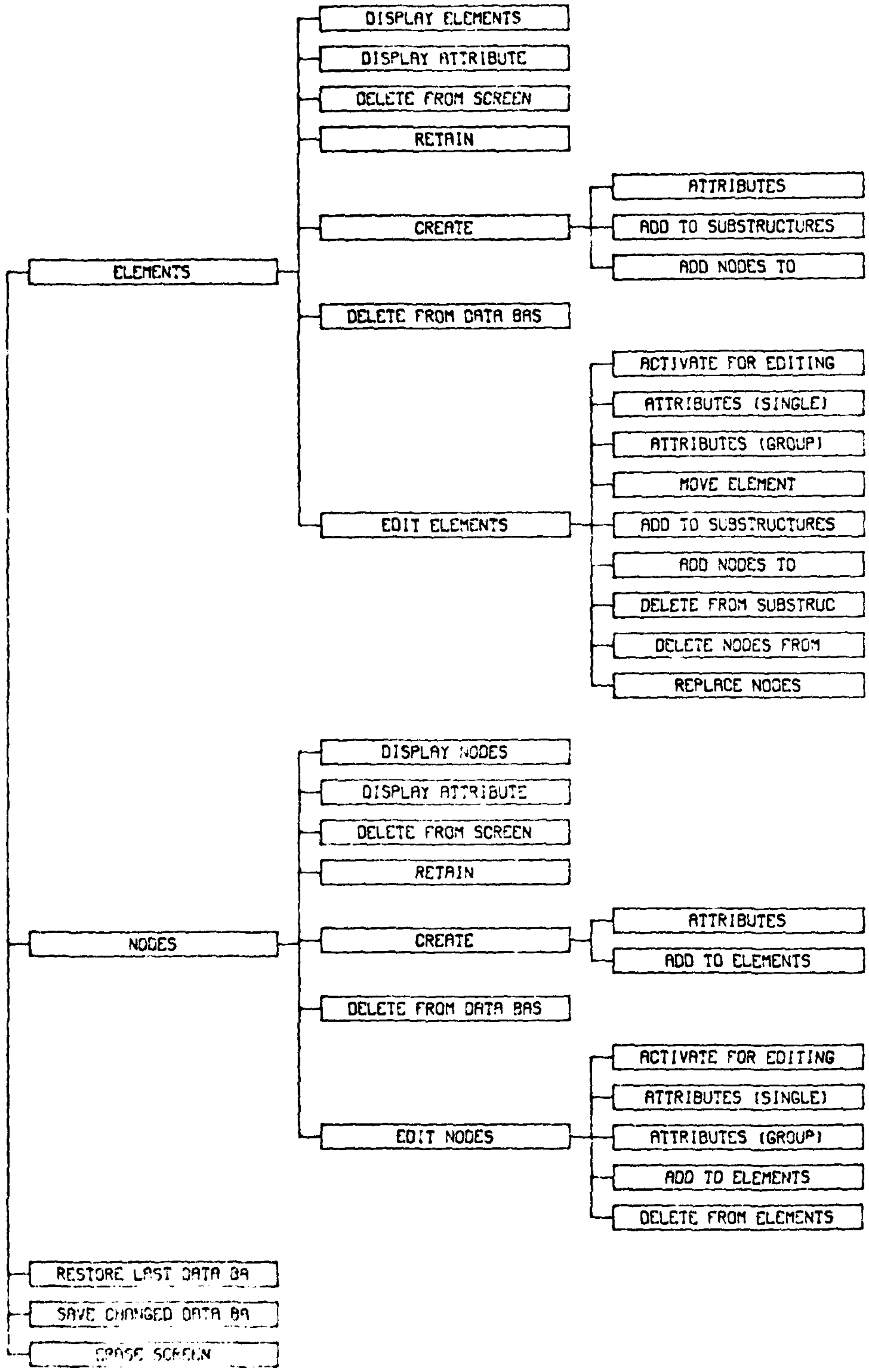


FIGURE 5.2.b. THE STAGING EDIT TREE

CHAPTER 6

ANALYSIS INTERFACE

STAGING interfaces with an analysis program through the use of conversion programs. In Chapter 2 of this manual, the user was introduced to the process of transferring analysis input data into the STAGING Data Base. This chapter describes how a user can retrieve data from a data base, and how he can extend the data base by adding the results of an analysis.

Also described in this chapter are the conversion programs available with STAGING. These conversion programs are used with NASTRAN, AXISOL, DOASIS, HONDO, FASTOP, and ADINA. Capabilities and limitations of the conversion programs are described and the batch and interactive commands required to execute these conversion programs are discussed.

6.1 Conversion Program 2

Once the user has corrected and optimized his model using STAGING, he is ready to execute an analysis program. Conversion Program 2 will be used to transfer information from the general data base to a card image file suitable for processing by the analysis program. Conversion Program 2 is the exact inverse of Conversion Program 1. The utility subroutines used are similar in appearance and structure.

6.1.1 NCINIT

NCINIT initializes the conversion program and defines the attributes to be retrieved from the data base. Attributes cannot be retrieved from the data base that have not been created. If a Conversion Program 1 has been used to create the data base, its initialization code can be used by Conversion Program 2. The order in which the user specifies the attributes is the order in which they are returned.

CALLING PARAMETERS:
CALL NCINIT (IATNOD, NATNOD, IATELM, NATELM)

Where: IATNOD - Node attribute array
NATNOD - Number of NODE attributes activated
IATELM - Element attribute array
NATELM - Number of ELEMENT attributes activated

6.1.2 NCSTR

NCSTR retrieves a structure and substructure linkage from the data base.

CALLING PARAMETERS:
CALL NCSTR (NUM, ATT, SUBS, NSUB)

Where: NUM - Structure name (40 characters or 4 words) NOTE: Each call to NCSTR will return a new structure name until the list has been exhausted, then NUM will be set to zero.

ATT - Dummy parameter

SUBS - Name of substructures attached to this structure (4 words per name)

NSUB - Number of substructure names supplied in SUBS

6.1.3 NCSUB

NCSUB retrieves a substructure and element linkage from the data base.

CALLING PARAMETERS:

CALL NCSUB (NUM, ATT, IELEM, NELEM)

Where: NUM - Substructure name (40 characters or 4 words) NOTE: Each call to NCSUB will return a new substructure name until the list has been exhausted, then NUM will be set to zero.

ATT - Dummy parameter

IELEM - Name of elements attached to this substructure (Integer format)

NELEM Number of element numbers supplied in IELEM

6.1.4 NCELEM

NCELEM retrieves elements, element attributes, and node linkages from the data base.

CALLING PARAMETERS:

CALL NCELEM (IELEM, ATT, NODE, NNODE)

Where: IELEM - Element number (integer format) NOTE: Each call to NCELEM will return a new element number until the list is exhausted, then IELEM will be set to zero. The element numbers are returned in ascending order.

ATT - Element attribute array

NODE - List of nodes attached to this element (integer format)

NNODE - Number of node numbers supplied in NODE.

6.1.5 NCNODE

NCNODE retrieves nodes and their attributes from the data base.

CALLING PARAMETERS:

CALL NCNODE (NODE, ATT)

Where: NODE - Node number (Integer format) NOTE: Each call to NCNODE will return a new NODE number until the list is exhausted, then NODE will be set to zero. The node numbers are returned in ascending order.

ATT - Node attribute array

6.1.6 NCASU

NCASU activates a substructure for retrieval. If the user has created several substructures and wishes to perform an analysis on specific substructures, NCASU supports that requirement. If NCASU is called, only the elements and nodes belonging to the listed substructures will be available to subsequent calls to NCELEM and NCNODE. If NCASU is not called, all information in the data base is available to the user. Note NCASU may be called at anytime after NCINIT and may be called as often as desired. However, only those substructures listed in the latest call will be active.

CALLING PARAMETERS:

CALL NCASU (SUBS, NSUBS)

Where: SUBS - Substructure names (40 characters or 4 words each)

NSUBS - Number of substructure names supplied in SUBS

6.1.7 NUMC

NUMC returns the number of items within a level. This utility function is useful in obtaining information about the data base.

CALLING PARAMETERS:

NUM = NUMC (ITYPE)

Where: ITYPE - data base level

= 1 - Structure

= 2 - Substructure

= 3 - Element

= 4 - Node

NUM - Number of items present in data base.

6.1.8 XNCTABG

Function XNCTABG is used to obtain block data stored by the subroutine CNTABL.

CALLING PARAMETERS:

VAL = XNCTABG (NAME,IR)

Where: NAME - 40-character name

IR - entry in array desired

VAL - data returned to user

6.1.9 NCFORM

Subroutine NCFORM is used to obtain the format stored for a specific block data table.

CALLING PARAMETERS:
CALL NCFORM (NAME,FORMAT)

Where: NAME - 40-character name
FORMAT - Format as supplied by user, up to 30 characters

6.1.10 NCTERM

NCTERM terminates the data conversion process. This must be the last NC routine called.

CALLING PARAMETERS:
CALL NCTERM

6.2 Conversion Program 3

This is the last conversion program of the analysis sequence. It is used to add the output results from an analysis run to an existing data base. Another set of conversion subroutines have been generated to aid in this task. They are similar in concept and function to those used in conversion programs one and two. At each level in the data base they are used to add or modify information in an existing data base.

6.2.1 NCINIT

NCINIT initializes the conversion subroutines and identify the attributes you wish to add or modify in the data base. This subroutine is the same one described in Section 2.4.1. Here the user specifies the order in which he will be supplying data. Only attributes arrays to be added to the data base need be activated here.

CALLING PARAMETERS:

CALL NCINIT (IATNOD, NATNOD, IATELM, NATELM)

Where: IATNOD - Node attribute array

NATNOD - Number of node attributes activated for program results data

IATELM - Element attribute array

NATELM - Number of element attributes activated for programs results data

6.2.2 NCSELM

NCSELM sets element attribute, values.

CALLING PARAMETERS:

CALL NCSELM (NUMBER, ATT)

Where: NUMBER - element number (integer format)

ATT - attribute array

6.2.3 NCSNOD

NCSNOD sets node attribute, values

CALLING PARAMETERS:

CALL NCSNOD (NUMBER,ATT)

Where: NUMBER - node number (integer format)

ATT - attribute array

6.2.4 NCTABS

NCTABS sets a value in a table.

CALL NCTABS (NAME,VAL,IR)

Where: NAME - the 40-character name of the table
VAL - the new value to insert into the array
IR - the location of the new value

6.2.5 NCTERM

NCTERM terminates the conversion subroutines and extend the STAGING Data Base.

CALLING PARAMETERS:
CALL NCTERM

6.3 Specific Analysis Conversion Programs

This section describes the availability and status of conversion programs for six analysis codes. These programs are for NASTRAN, FASTOP, AXISOL, DOASIS, HONDO and ADINA. The limitations and capabilities of each of these conversion programs are described. The user can extend or modify the capabilities of these programs by using the facilities described in Chapter 2 and Chapter 6 of this manual and consulting the STAGING manual. In all of these program descriptions "TAPE0" is used to signify TAPE Number Zero.

6.3.1 NASTRAN/STAGING Interface (NASCON1)

The NASTRAN system has been developed by NASA over a number of years as a very powerful general purpose three-dimensional finite element analysis system. Its capabilities have been well publicized and, in fact, the specific capabilities depend on the so-called level of development in NASTRAN. Therefore, input data formats depend on a development level. Hence, the data conversion routines must also be written for a specific level. The NASTRAN documentation consists of an excellent User's Guide, and a Theoretical Manual. [1,2]

Data conversion routines have been written for NASTRAN Level 17. NASCON1 reads a NASTRAN data deck, extracts the node and element cards, and inserts these in the STAGING Data Base. NASCON2 generates the correct

NASTRAN data deck from a STAGING Data Base created by NASCON1. NASCON3 extracts the static displacements, eigenvectors, and element stresses from the NASTRAN punch file and stores them in the data base. The element stresses are stored in the STAGING Data Base in the same coordinate system in which they were produced by NASTRAN. No rotation or translation is made of the element stress components.

6.3.1.1 How to Execute NASCON1. NASCON1 can be executed in either a batch mode or using INTERCOM. Depending on the size and type of NASTRAN model, the user should allow sufficient central processing time. The following commands or job control cards are required:

a. ATTACH, ABS, NASTRANCONVERSIONONEABS, ID = STAGING
   This card attaches the absolute version of NASCON1.
b. ATTACH, TAPE1, pfn, id = user id.
   Tape 1 is the user's NASTRAN data deck, pfn and user id are the appropriate permanent file and id.
c. ABS. - Execute the NASCON1 program.
d. CATALOG, TAPEO, pfn, id = user id.
   TAPEO is the user's STAGING data base. Execution of NASCON1 automatically requests TAPEO as a permanent file.

6.3.1.2 How to Execute NASCON2. NASCON2 can be executed in either a batch mode or using INTERCOM. The following commands or job control cards are required:

a. ATTACH,XX,pfn, id = user id
   Logical file XX is the user's STAGING data base. In order for NASCON2 to work correctly, this data base will have to be originally created by NASCON1 and, if necessary, subsequently edited using STAGING.
b. COPY (XX, TAPEO)
c. ATTACH,ABS,NASTRANCONVERSIONTWOABS,ID = STAGING.
d. ABS.
e. NASCON2 creates a file with lfn = TAPE1. This is the NASTRAN data deck and the user can either catalog it or dispose it to the punch file.

6.3.1.3 How to Execute NASCON3. This program adds displacements, mode shapes and stresses to the user's STAGING data base. In order to use this program, the user must have the following control cards during a NASTRAN analysis.

DISP (PRINT,PUNCH) = ALL
ELSTRESS (PRINT,PUNCH) = ALL

The user must also catalog the punch file which will later be used by NASCON3. The NASCON3 program can be run either in a batch mode or interactively using INTERCOM.

The following intercom commands or control cards (in a batch submission) will convert the NASTRAN output in the user's data base originally created by NASCON1).

a. ATTACH,TAPE1,pfn,id = user id.
   TAPE1 is the NASTRAN punch file created during the execution of NASTRAN
b. ATTACH,XX,pfn,id = user id.
   Logical file XX is the user's STAGING data base. In order for NASCON3 to work correctly, this data base will have to be originally created by NASCON1 and subsequently edited using STAGING.
c. REQUEST,TAPEO,*pf.
d. COPY (XX,TAPEO)
e. ATTACH,ABS,NASTRANCONVERSIONTHREEABS, ID = STAGING.
f. ABS.
g. CATALOG,TAPEO,pfn,id= user id.
   TAPEO is the extended user's STAGING data base. It has the NASTRAN analysis displacements and stresses.

6.3.1.5 Capabilities and Limitations of NASCON1. The following list describes the options available in NASCON1 and also details the limitations of NASCON1.

a. The bulk data card format must be of the small field type, i.e., cards using 8 column fields.
b. Except for the GRID, MAT1, MATT1 bulk data cards, and bulk

cards defining element connectivity, the rest of the bulk cards are stored in the data base as tables and as such are not accessible during a STAGING session.

c. The GRID card must define the location of the point in a global coordinate system.

d. The GRID card should only contain the grid number and the location (X1, X2, and X3) of the point. Other information such as permanent constraints is not interpreted and is lost.

6.3.2 AXISOL/STAGING Interface

Program AXISOL is a two dimensional finite-element thermal stress analysis program which allows for orthotropic material properties which may be temperature dependent. Program AXISOL includes a provision for a multilinear stress-strain curve. Documentation for AXISOL consists of a fairly adequate user's guide describing input data for the program. [3]

Data conversion routines for AXISOL consist of AXICON1, AXICON2, and AXICON3. AXICON1 allows the AXISOL program input deck to be read and the nodes and element tables used to create a STAGING Data Base. AXICON2 creates an AXISOL data deck from the STAGING Data Base. AXICON3 reads and stores both the AXISOL computed stresses and displacements in the STAGING Data Base.

6.3.2.1 How to Execute AXICON1. AXICON1 can be executed in either a batch mode or using INTERCOM. Depending on the size and type of AXISOL model, the user should allow sufficient central processing time. The following commands or job control cards are required:

a. ATTACH,ABS,AXISOLCONVERSIONONEABS, ID = STAGING
   This card attaches the absolute version of AXICON1.

b. ATTACH,TAPE1, pfn, id = user id.
   TAPE1 is the user's AXISOL data deck, pfn and user id are the appropriate permanent file and id.

c. ABS.

d. CATALOG,TAPEO,pfn,id = user id.
   TAPEO is the user's STAGING Data Base. Execution of AXICON1

automatically requests TAPEO as a permanent file.

6.3.2.2 How to Execute AXICON2. AXICON2 can be executed in either a batch mode or using INTERCOM. The following commands or job control cards are required:

a. ATTACH,XX,pfn,id = user id.
   Logical file XX is the STAGING Data Base. In order for AXICON2 to work correctly this data base will have to be originally created by AXICON1 and subsequently edited using STAGING.
b. COPYBF (XX,TAPEO)
c. ATTACH,ABS, AXISOLCONVERSIONTWOABS,ID = STAGING.
   This cards attaches the absolute version of AXICON2.
d. ABS.
e. AXICON2 creates a file with lfn = TAPE1. This is the AXISOL data deck and the user can either catalog it or dispose it to a punch file.

6.3.2.3 How to Execute AXICON3. AXICON3 can be executed in either a batch mode or using INTERCOMS. The following commands or job control cards are required:

a. ATTACH,TAPE1,pfn,id = user id.
   TAPE1 is the IAXISOL file created during the execution of AXISOL (special version).
b. ATTACH,XX,pfn,id = user id.
   Logical file XX is the user's STAGING data base. In order for AXICON3 to work correctly this data base must have been created by AXICON and may have been subsequently edited using STAGING.
c. REQUEST,TAPEO,*pf.
d. COPYBF (XX,TAPEO)
e. ATTACH,ABS,AXISOLCONVERSIONTHREEABS,ID = STAGING.
f. ABS
g. CATALOG,TAPEO,pfn,id = user id.
   TAPEO is the extended user's STAGING Data Base. It has the AXISOL analysis displacements and stresses stored.

6.3.2.4 Capabilities and Limitations of AXICON. The AXISOL conversion routines can only be used for data prepared for an elastic version of AXISOL (AXISO2). Also in order to execute AXICON3 several changes were made to the basic elastic version of AXISOL. These changes create a file on which the displacements and stresses are stored in a format compatible with AXICON3.

The changes are:

1. In the main program the displacements for each node were stored on file 20 (TAPE20) using a format of (I5, 2E15.6). The node number is also stored.
2. In the subroutine STRESS, the stress tensors (six components) were stored on file 20 (TAPE20) using a format of (I5, 6E15.6). The element number is also stored.

6.3.3 DOASIS/STAGING Interface

Program DOASIS is a two dimensional finite-element thermal stress analysis program. DOASIS allows for the analysis of plane stress, plane strain or axisymmetric problems. It allows for temperature-dependent orthotropic material behavior with bilinear stress-strain curves. Up to 40 material/temperature combinations are permitted. (The number of materials times the number of temperatures is less than 40.) Documentation for DOASIS is given in Air Force Materials Laboratory Report No. TR75-37 in three volumes. Volume 1 is a theoretical and programmer's manual for DOASIS. Volume 2 is a programmer's manual for the preprocessors and post-processors; Volume 3 is the user's manual for all of the DOASIS system. [4-6]

Data conversion routines were developed to communicate directly with DOASIS. This means that they will not communicate with pre and post processors of DOASIS. If the mesh generator, MESGEN, or temperature input generator, TEMINT, of DOASIS are used, they must be triggered to punch out DOASIS input data card decks. The first data conversion routine, DOASC01, reads the DOASIS input data deck, selects from it the nodal and element information, and stores this in the STAGING Data Base. DOASC01 reads the temperatures, either for nodes or elements, if they were given on the

corresponding nodal or element card or if they are given on separate cards. DOASCO2 creates a DOASIS input data deck from the corrected data base and is basically a mirror of the DOASCO1. DOASCO3 reads the output stresses and the output displacements, and adds these to the STAGING data base. Stresses are stored as element attributes. Thay are also interpolated to the node points and stored as Z-displacements. In this way, the stress distributions can be viewed as displaced surface plots relative to the plane of the problem. Temperatures are also stored in a Z-displacement format so that they may be viewed in a similar manner.

6.3.3.1 How to Execute DOASCO1. DOASCO1 can be executed in either a batch mode or using INTERCOM. Depending on the size and type of DOASIS model, the user should allow sufficient control processing time. The following commands or job control cards are required:

a. ATTACH,ABS,DOASISCONV1ABS,ID=STAGING

This card attaches the absolute version of DOASCO1.

b. ATTACH,TAPE1,pfn,id=user id.

Tapel is the user's DOASIS data deck (patrol), pfn and user id. are the appropriate permanent file name and id.

c. ABS - Execute the DOASCO1 program.

d. CATALOG,TAPEO,pfn,id = user id.

TAPEO is the users STAGING data base. Execution of DOASCO1 automatically requests TAPEO as a permanant file.

6.3.3.2 How to Execute DOASCO2. DOASCO2 can be executed in either a batch mode or using INTERCOM. The following commands or job control cards are required:

a. ATTACH,XX,pfn,id = user id

Logical file xx is the user's STAGING data base. In order for DOASCO2 to work correctly, this data base will have to be originally created by DOASCO2 and, if necessary, subsequently edited using STAGING.

b. COPY (XX, TAPEO)

c. ATTACH,ABS,DOASISCONV2ABS,ID = STAGING.

d. ABS.

e. DOASCO2 creates a file with lfn = TAPE1. This is the DOASIS

data deck and the user can either catalog it or dispose it to the punch file.

6.3.3.3 How to Execute DOASCO3. This program adds displacements, and stresses to the user's STAGING data base. In order to use this program, the user must set the IOSAVE equal to 3 and catalog the output file (tape 4), to be input to DOASCO3.

The following intercom commands or control cards (in a batch submission) will insert the DOASIS output into the user's data base (originally) created by DOASCO1.

a. ATTACH,TAPE4,pfn,id = user id.
   TAPE4 is the DOASIS output file created during the execution of DOASIS
b. ATTACH,XX,pfn,id= user id.
   Logical file XX is the user's STAGING Data Base. In order for DOASCO3 to work correctly, this data base will have to be originally created by DOASCO1 and subsequently edited using STAGING.
c. REQUEST,TAPEO,*pf.
d. COPYBF (XX,TAPEO)
e. ATTACH,ABS,DOASISCONV3ABS, ID = STAGING.
f. ABS.
g. CATALOG,TAPEO,pfn,id = user id.
   TAPEO is the extended user's STAGING data base. It has the DOASIS analysis displacements and stresses.

6.3.3.4 Capabilities and Limitations of DOASCON2. DOASCO1 and DOASCON2 will accept any type of input model and boundary condition set option for DOASIS. This includes the option of having temperatures specified on the node or element cards or on seperate card sets. As noted above, DOASCO1 or DOASCO2 interfaces with the card input option of DOASIS. If DOASIS generators are used, they must output data in card formats for DOASIS.

DOASCO3 will convert each of the element stresses and nodal displacements into data base format for adding to the DOASCO1 created model. In addition DOASCO3 interpolates each of the element stresses to

nodal stresses and stores these as pseudo out-of-plane displacements, this allows the stresses to be studied in surface carpet plot formats for easier understanding. These stresses may be retrieved by invoking the time step option in the DISPLAY RESULTS section of STAGING. The displacement time step components are as follows:

Time Step	Component
0, 1	In-Plane Displacements
2	Normal x stress
3	Normal y stress
4	Normal z stress
5	Shear xy stress
6	Effective stress ($\bar{\sigma}$)
7	Effective strain ($\bar{\varepsilon}$)
8	Nodal temperature

Proper scaling function for each of these components can be interactively selected for the most meaningful display.

6.3.4 HONDO/STAGING Interface

Program HONDO was developed for the analysis of impact problems in which the strains may be as high as 30-40%. HONDO uses an explicit time integration scheme to solve the problems. It allows for two-dimensional plane strain or axisymmetric deformation. Input deck formats resemble quite closely both the AXISOL and the DOASIS programs since it was derived from the original stress codes created by Wilson. There is a very good theoretical and user's document describing the HONDO code. [7]

Data conversion routines have been written for the HONDO program. Conversion Routine (HONCON1) reads HONDO input data, extracts the nodes and element tables and stores these in the STAGING Data Base so that the user can look at the geometric model. Conversion Routine 3 (HONCON3) captures the stresses and displacements at preselected times in the dynamic impact event and stores these in the STAGING Data Base as subscripted displacements or subscripted stresses for the nodes and

elements respectively. (*No HONCON2 conversion routine* has been written to transform a STAGING Data Base into a HONDO input deck.)

6.3.4.1 How to Execute HONCON1. HONCON1 can be executed in either a batch mode or using INTERCOM. Depending on the size and type of HONDO model, the user should allow sufficient computer processing time. The followng commands or job control cards are required:

a. ATTACH,ABS,HONDOCONV1ABS, ID = STAGING
   This card attaches the absolute version of HONCON1.
b. ATTACH,TAPE1,pfn, id = user id.
   Tape1 is the user's HONDO data deck, pfn and user id are the appropriate permanent file and id.
c. ABS. - Execute the HONCON1 program.
d. CATALOG,TAPE0,pfn, id = user id.
   TAPE is the user's STAGING data base. Execution of HONCON1 automatically requests TAPE0 as a permanent file.

6.3.4.2 How to Execute HONCON3. HONCON3 adds time dependent displacements to the STAGING Data Base. Multiple time steps may be outputted on the HONDO plot file and drawn sequentially.

The following intercom commands or control cards convert the HONDO output in the user's data base (originally) created by HONCON1.

a. ATTACH,TAPE1,pfn,id = user id.
   TAPE1 is the HONDO plot file created during the execution of HONDO.
b. ATTACH,XX,pfn,id = user id.
   Logical file XX is the user's STAGING data base. In order for HONCON3 to work correctly, this data base will have to be originally created by HONCON1 and subsequently edited using STAGING.
c. REQUEST,TAPE0,*pf.
d. COPYBF (XX,TAPE0)
e. ATTACH,ABS,HONDOCONV3ABS(ID = STAGING)
f. ABS.
g. CATALOG,TAPE0,pfn,id = user id.

TAPEO is the extended user's STAGING data base. It has the HONDO analysis displacements.

6.3.4.3 Capabilities and Limitations of HONDO Conversion Routines. As noted earlier, the HONDO conversion routine set does not at present include HONCON2. This routine could easily be created from AXICON2, if needed, because of the similarity in input data formats between the two codes.

HONCON1 allows for the various material input options for HONDO and creation of either triangular or quadvilations elements.

HONCON3 reads the stress outputs but does not store these in the STAGING data base, at present. This capability can be added, if needed. Caution is advised to insure that HONDO3 is interfaced properly with the output plot file of HONDO since a number of versions of this code are in existence.

6.3.5 FASTOP/STAGING Interfaces

The FASTOP program was developed by Grumman for the optimization by flutter and by strain of an aircraft structure. Adequate documentation of the FASTOP system is given in AFFDL report TR75-137, Volumes 1 and 2. Volume 1 is a theory and applications report while Volume 2 is a program user's manual. [8,9]

Five data conversion routines have been written for interfacing between the FASTOP system and the STAGING Data Base. The first data conversion routine (SOPCON1) accepts a SOP input, extracts from it a number of geometric entities and stores these in the STAGING Data Base. Data conversion routine 2 (SOPCON2) creates a corrected SOP input data deck from the data in the STAGING Data Base. SOPCON3 stores the output results from the SOP program in the data base. Data conversion routine 4 (FOPCON1) accepts a FOP input data deck and adds it to the STAGING Data Base created by SOPCON3 from the SOP output. Data conversion routine 5 (FOPCON3) extracts certain data from the output of the FOP analysis and stores it in the STAGING Data Base for display.

6.3.6 ADINA/STAGING Interface

Program ADINA is a general purpose two-and three-dimensional finite element analysis program developed by Professor Bathe. ADINA is an extension of the program NONSAP, on which Professor Bathe had worked earlier. Documentation for the ADINA program is quite good and, in contrast to other proprietary finite-element codes, and ADINA user can obtain the ADINA source code when he joins the ADINA user's group. [10]

The ADINA data conversion programs are ADICON1, ADICON2, and ADICON3. (Note, only ADICON1 has been developed.)

ADICON1 converts ADINA input data deck into a STAGING Data Base. The input to ADICON1 consists of:

1. First card - NUMNP,NUMEL,ND

   NUMNP - Number of Nodes in Model

   NUMEL - Number of Elements in Model

   ND - Flag for indicating element type

   (e.g. 2 - 2D elements)

   3 - 3D elements

2. Nodal coordinate cards (ADINA)

   - All nodes must be present
   - X,Y,Z coordinates are assummed global rectangular

3. Element connectivity cards (ADINA)

   2 cards if No. equals 2

   3 cards if No. equals 3

6.3.6.1 How to Execute ADICON1. ADICON1 can be executed in either a batch mode or using INTERCOM. Depending on the size and type of ADINA model, the user should allow sufficient control processing time. The following commands or job control cards are required:

a. ATTACH,ABS,ADINACONVERSIONONEABS,ID=STAGING.

This card attaches the absolute version of ADICON1.

b. ATTACH,TAPE1,pfn,id=user id.

Tapel is the user's ADINA data deck (patrol), pfn and user id. are the appropriate permanent file and id.

c. ABS.

d. CATALOG,TAPEO,pfn,id=user id.

TAPEO is the user's STAGING Data Base. Execution of ADICON1 automatically requests TAPEO as a permament file.

6.3.6.2 How to Execute ADICON2. This conversion program is not available.

6.3.6.3 How to Execute ADICON3. This conversion program is not available.

6.3.6.4 Capabilities and Limitations of ADICON. Only a limited ADINA interface is available. ADICON1 reads nodes and elements cards into the STAGING Data Base for displaying and editing.

6.4 SUMMARY

The data conversion routines in this chapter provide the STAGING user with the capability of graphical interaction with a number of analysis codes. In addition, the user, by following the procedures outlined in this manual and the companion VOLUME III, can construct data conversion routines for his own favorite program. This facility of STAGING will make it broadly useful in the engineering community.

Figure 6.1 is a listing of procedure CONVGEN which allows creation of the absolute conversion routine programs from the source language programs on the CONVERSIONPL.

```
PROC,CONVGEN,PART1=,PART2=.

   THIS PROCEDURE WILL CREATE AN ABSOLUTE FOR
   ANY ONE OF THE ANALYSIS CONVERSION PROGRAMS
   AVAILABLE WITH STAGING.

   FOR THE "PART1" PARAMETER, THE FOLLOWING SYMBOLS
   ARE VALID

     ADICON   FOR ADINA ANALYSIS CONVERSION PROGRAM
     ANSCON   FOR ANSYS ANALYSIS CONVERSION PROGRAM
     AXICON   FOR AXISOL ANALYSIS CONVERSION PROGRAM
     DOASCO   FOR DOASIS ANALYSYS CONVERSION PROGRAM
     FOPCON   FOR FASTOP ANALYSIS CONVERSION PROGRAM
     HNDO     FOR HONDO ANALYSIS CONVERSION PROGRAM
     NASCON   FOR NASTRAN ANALYSIS CONVERSION PROGRAM
     SOPCON   FOR FASTOP ANALYSIS CONVERSION PROGRAM

   FOR THE "PART2" PARAMETER, THE FOLLOWING SYSMBOLS
   ARE VALID

     1 : CONVERSION PROGRAM 1
     2 : CONVERSION PROGRAM 2
     3 : CONVERSION PROGRAM 3

   THE USE OF THIS PROCEDURE REQUIRES THE PRESENCE OF A
   LOCAL FILE NAMED "UPIN", WHICH WOULD CONTAIN THE UPDATE
   DIRECTIVES TO PULL THE SOURCE PROGRAM OFF THE UPDATE PL.
   THE NAME OF THE UPDATE DECK FOLLOWS THE ABOVE SYMBOLIC
   NAMING RULES (I.E., PART1 CONCATENATED WITH PART2).

   FOR EXAMPLE, IF YOU WANT TO CREATE AN ABSOLUTE FOR
   SOPCON1, THEN THE FOLLOWING COMMAND WOULD DO THE
   TRICK

       BEGIN,CONVGEN,PART1=SOPCON,PART2=1.

RETURN,OLDPL,ABS,LIB.
ATTACH,OLDPL,CONVERSIONPL,ID=STAGING3.
REWIND,UPIN.
UPDATE(Q,I=UPIN,L=1)
FIN(Z,I=COMPILE,R=0)
REWIND,LGO.
REQUEST,ABS,*PF.
ATTACH,LIB,SUPPORTLIB,ID=STAGING3.
LIBRARY(LIB)
LDSET(PRESET=ZERO)
LOAD(LGO)
NOGO(ABS)
CATALOG,ABS,PARTL_PART2_ABS,ID=STAGING3,RP=999.
REVERT.
```

FIGURE 6.1. PROCEDURE FOR CREATING ABSOLUTE CONVERSION PROGRAMS.

CHAPTER 7

POSTPROCESSING ANALYSIS RESULTS

Once the user has executed a finite element analysis and appended the results of his analysis to a STAGING Data Base, he can use STAGING to perform several postprocessing tasks.

The Postprocessing module of STAGING performs additional tasks, allowing a user to obtain additional engineering data that his finite element analysis may not provide. Section 7.1 discusses two capabilities for creating stresses. Once the new data has been created, the user can then use the more common postprocessing functions to view the new data.

The more common of the postprocessing tasks, such as deformed plots, contours, and XY plots are discussed in detail in Section 7.2, 7.3, and 7.4. These tasks basically display data already in the data base in various formats.

7.1 CREATE DATA

The Postprocessing module creates two new data items; namely, the principal stresses and the equivalent stresses. It assumes that the STAGING Data Base contains the following defined element attributes:

X N STRESS
Y N STRESS
Z N STRESS
X S STRESS
Y S STRESS
Z S STRESS

7.1.1 PRINCIPAL STRESSES

If the user picks PRINCIPAL STRESSES after choosing CREATE DATA, the system calculates and stores element attribute values for:

MAX STRESS - Maximum Principal Stress
INT STRESS - Intermediate Principal Stress
MIN STRESS - Minimum Principal Stress

7.1.2 EFFECTIVE STRESSES

If the user picks EFFECTIVE STRESS after choosing CREATE DATA, the system calculates and stores the following element attribute value:

EQU STRESS - Effective Stress

CREATE DATA creates a new item in the data base for **all** elements in the data base regardless of the type of model. Hence, in two dimensional problems, the user has to supply zero values in appropriate stress attributes. The system cannot create the above data for multiple time or load dependent stresses; only one load or one step (X N STRESS, Y N STRESS, and Z N STRESS, for example) can be created.

7.2 DEFORMED PLOTS

DEFORMED PLOT applies a displacement to the node points in a model. The displacement can be the result of a load condition, a mode shape, or a time step in a dynamic analysis. There are currently 999 steps, numbered from 1 to 999 available for a model. The user can select DEFORMED PLOT, set a scale factor and the step to be looked at, and pick DRAW DEFORMED PICTURE to draw the model in 2D or 3D. A dashed line plot is drawn to reduce confusion between deformed and undeformed plots. If the deformed plot is to be superimposed on top of the undeformed plot, the original must be displayed before selecting DEFORMED PLOT. One then enters DEFORMED PLOT mode, sets the proper values, and picks DRAW DEFORMED PICTURE. Deformed mode stays in effect until SET UNDEFORMED MODE is selected. Figure 7.1 shows a typical deformed plot.

7.2.1 SET UNDEFORMED MODE

This function resets the model drawing line style to solid lines. In order to distinguish between normal models and deformed plots, the latter are drawn with dashed lines. In addition, any new drawings are also done with dashed line, unless SET UNDEFORMED MODE is selected before leaving DEFORMED PLOT. When this button is picked, the system returns to the previous menu; that is, it also executes a RETURN function.

7.2.2 SET SCALE FACTOR

This function allows the user to specify a scaling of the nodal displacements. The scale factor is specified by entering a data value via the keyboard. The default scale factor is one (1.0). SET SCALE FACTOR is used to exaggerate the displacement as shown in Figure 7.1.

7.2.3 BY LOAD CONDITION

BY LOAD CONDITION selects the displacement values from attributes X LOAD, Y LOAD, and Z LOAD.

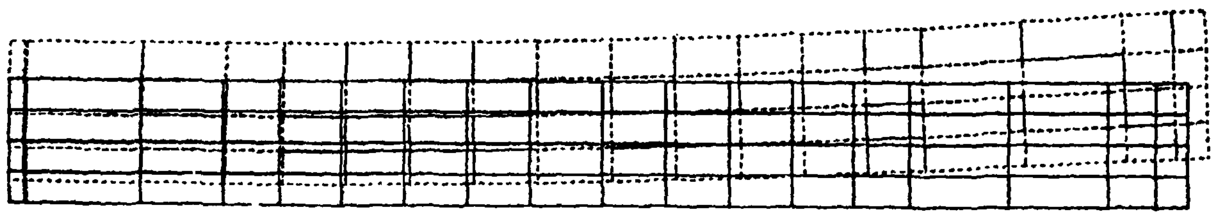


FIGURE 7.1. A SIMPLE DEFORMED PLOT

7.2.4 BY MODE SHAPE

BY MODE SHAPE selects the displacement values from X MODE, Y MODE, and Z MODE.

7.2.5 BY TIME STEP

BY TIME STEP selects the displacement values from attributes X DISP, Y DISP, and Z DISP. Furthermore, it allows the user to specify, via the keyboard, which of the time step displacements is to be plotted. This value may be from 1 to 999. (Note that 0 and 1 are equivalent.) The default time step is 1.

7.2.6 DRAW DEFORMED PICTURE

When DRAW DEFORMED PICTURE is selected, the system draws the deformed plot according to the scale factor and displacement values selected by the user. If the undeformed model is displayed, the deformed plot is superimposed on this model.

7.3 X-Y PLOTS

The X-Y PLOT function provides the capability to plot pairs of element or node attribute values on two axis graphs. Using X-Y PLOT, an attribute can be drawn versus any other attribute for any part of the data base. The method of specifying active elements or nodes is the same as displaying the model. If the X-Y PLOT button is picked, those items already active will remain active. Additional items on any of the four levels can be activated.

The construction of the X-Y plot allows the choice of up to ten attribute pairs for plotting. The first plot controls the size and limits of the plotting space and subsequent curves are plotted in the same style and scale as the first. The first curve is drawn automatically, while the user must request the drawing of subsequent curves. All values on the x-axis of the plots are sorted in increasing numeric order. The corresponding Y values remain associated with the X values as would be

expected. Each of the axes can be either a linear scale or a logarithmic scale. The user can select from different line styles to distinguish individual curves.

The picture modifying capability such as split screen and zooming can be used on plots. Figure 7.2 illustrates some of the flexibility of the X-Y PLOT function.

7.3.1 ACTIVATE DATA

This function activates data base items for plotting, in addition to any items already active. This function is available under both ELEMENTS and NODES.

7.3.2 X-AXIS

The X-AXIS function specifies which attribute values are to be used as the x-axis values.

7.3.3 Y-AXIS

The Y-AXIS function specifies which attribute values are to be used as the y-axis values.

7.3.4 CHANGE GRAPH

The CHANGE GRAPH function provides additional capabilities for modifying the plotting values. In particular, it provides control over rescaling the graph to the largest curve (as opposed to the first curve), the scaling along each axis (linear or logarithmic), titling, minimum and maximum axis values, and the order of sorting the data values.

7.3.4.1 REPLOT GRAPH
7.3.4.2 RESCALE GRAPH
7.3.4.3 GRAPH STYLE
7.3.4.4 MINS AND MAXES
7.3.4.5 CHANGE SORT

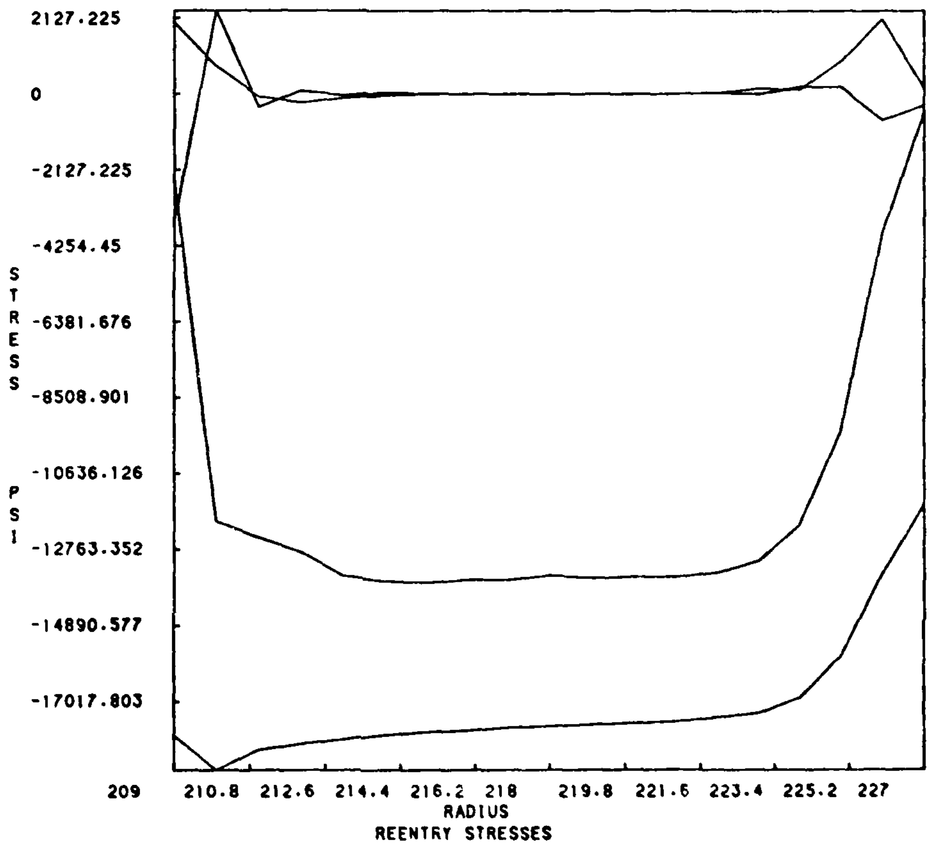


FIGURE 7.2. MULTIPLE X-Y PLOT

7.3.5 CHANGE LINE

The CHANGE LINE function allows the user to select different line styles (connected points. solid, or dashed), to delete selected lines, and to specify offsets and scale factors for each curve.

7.3.5.1 LINE STYLE
7.3.5.2 REPEAT FACTOR
7.3.5.3 DELETE LINE
7.3.5.4 X SCALE/OFFSET
7.3.5.5 Y SCALE/OFFSET

7.3.6 ERASE

The ERASE function erases the screen. This is used before REPLOT or RESCALE since neither of those functions erase the screen before drawing the next plot.

7.3.7 DEACTIVATE

The DEACTIVATE function deactivates **all** active nodes and elements. Active structures and substructures remain active.

7.4 CONTOUR PLOT

The CONTOUR PLOT function generates contour plots for any collection of elements or nodes. The contour lines are always projected onto a principal plane (such as the X-Y plane). The user can specify the number of contour lines and whether or not they are to be labeled. The structure (or collection of elements or nodes) being contoured, can be displayed with the contour lines on top. Alternatively, the user may display only the convex hull created by the active items.

The user can select either ELEMENTS or NODES to specify which attribute value is to be contoured. When ELEMENTS are contoured, the system computes the centroids of the elements and interpolates between

these points. For NODES, the actual nodal coordinates are used for contouring. For best results contour should be used with convex sets of elements. Figure 7.3 shows an example of contour plots.

7.4.1 ACTIVATE DATA

The user identifies data to be contoured in the same manner as items are identified for display (See Section 3.1).

7.4.2 SELECT ATTRIBUTE

After the user activates the data, he identifies the attribute value to be contoured with SELECT ATTRIBUTE.

7.4.3 DRAW PICTURE

After the user has activated elements and selected an attribute value to be contoured, the DRAW PICTURE function will display the contour plot.

7.4.4 CHANGE TITLES

The CHANGE TITLES function allows the user to specify a title for the contour plot. When CHANGE TITLES is selected the system displays input template and requests that the user enter a new title. If the user wishes to have a blank title, he must enter blanks for all input data parameters which have values.

7.4.5 CHANGE PARAMETERS

The CHANGE PARAMETERS function allows the user to control certain of the contouring parameters such as the minimum and maximum values, the contour interval and labeling.

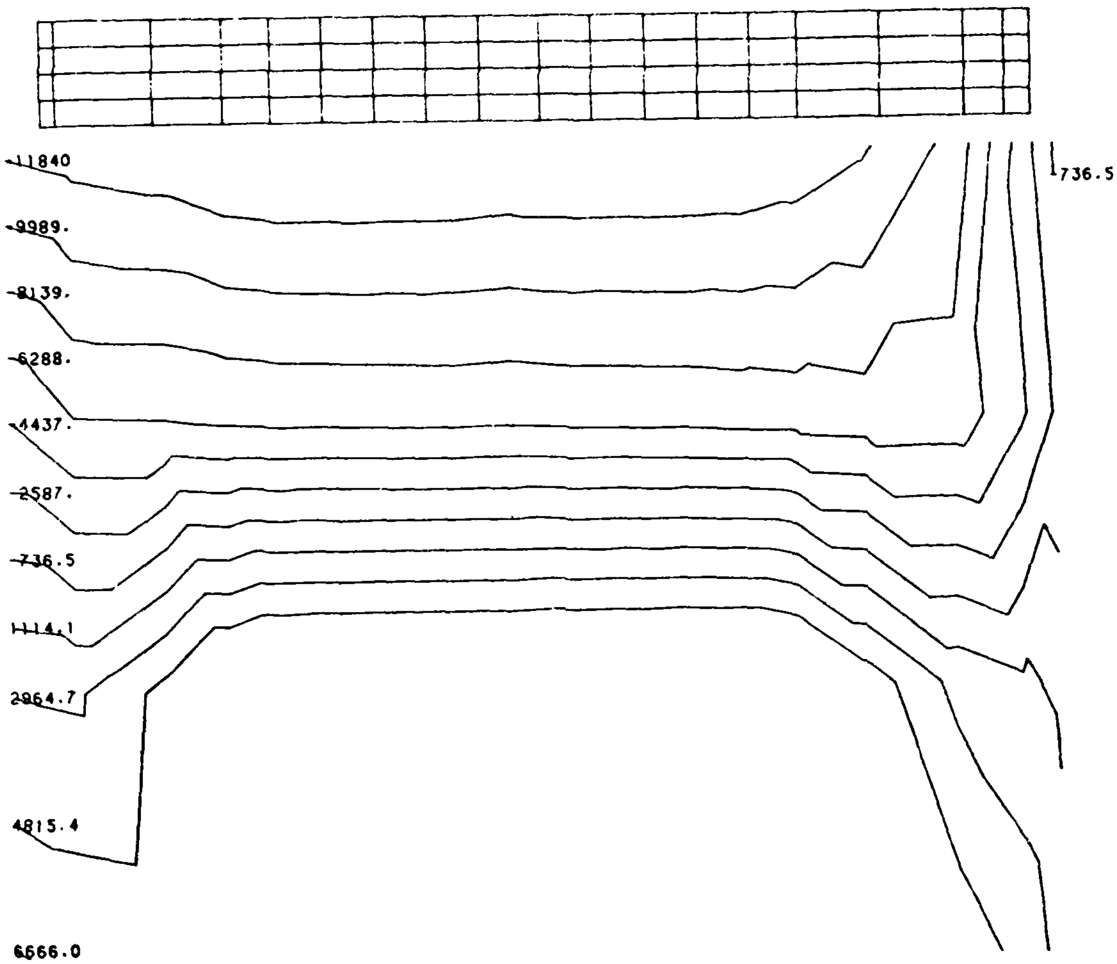


FIGURE 7.3. A TYPICAL CONTOUR PLOT

7.4.6 ERASE

The ERASE function erases the screen.

7.4.7 DEACTIVATE

The DEACTIVATE function deactivates **all** active nodes and elements. Active structures and substructures remain active.

7.5 SUMMARY

An engineer must study and understand deeply the results of a finite element analysis to use it wisely in achieving a successful design. STAGING provides the user with the capability to display the analysis results and to interactively manipulate this display in many ways. This capability allows the user to quickly and easily study his results until he really understands them and use this understanding to improve his design.

Figure 7.4 illustrates the STAGING command trees that carry out the postprocessing activities in STAGING.

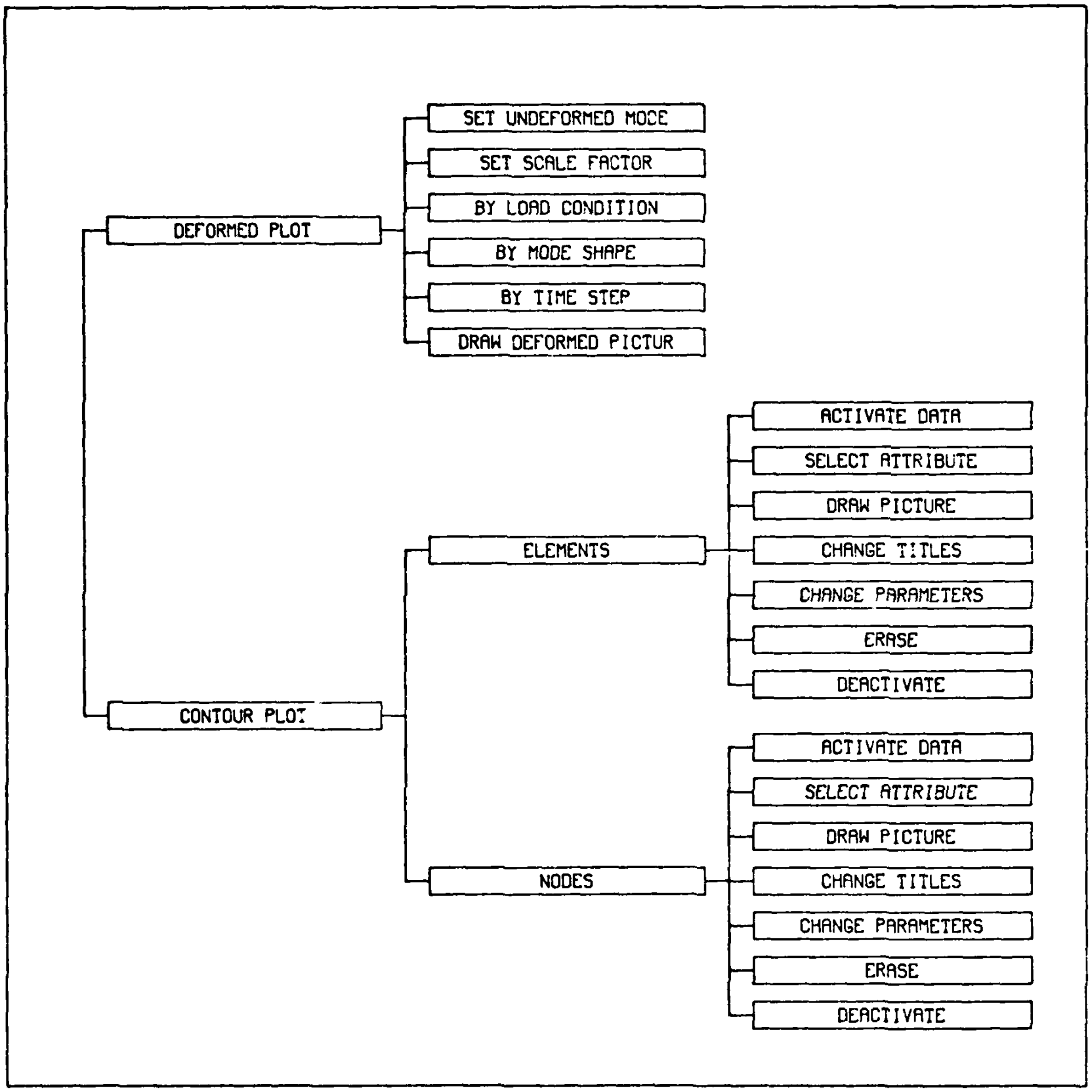


FIGURE 7.4.a. THE STAGING POSTPROCESSOR TREE

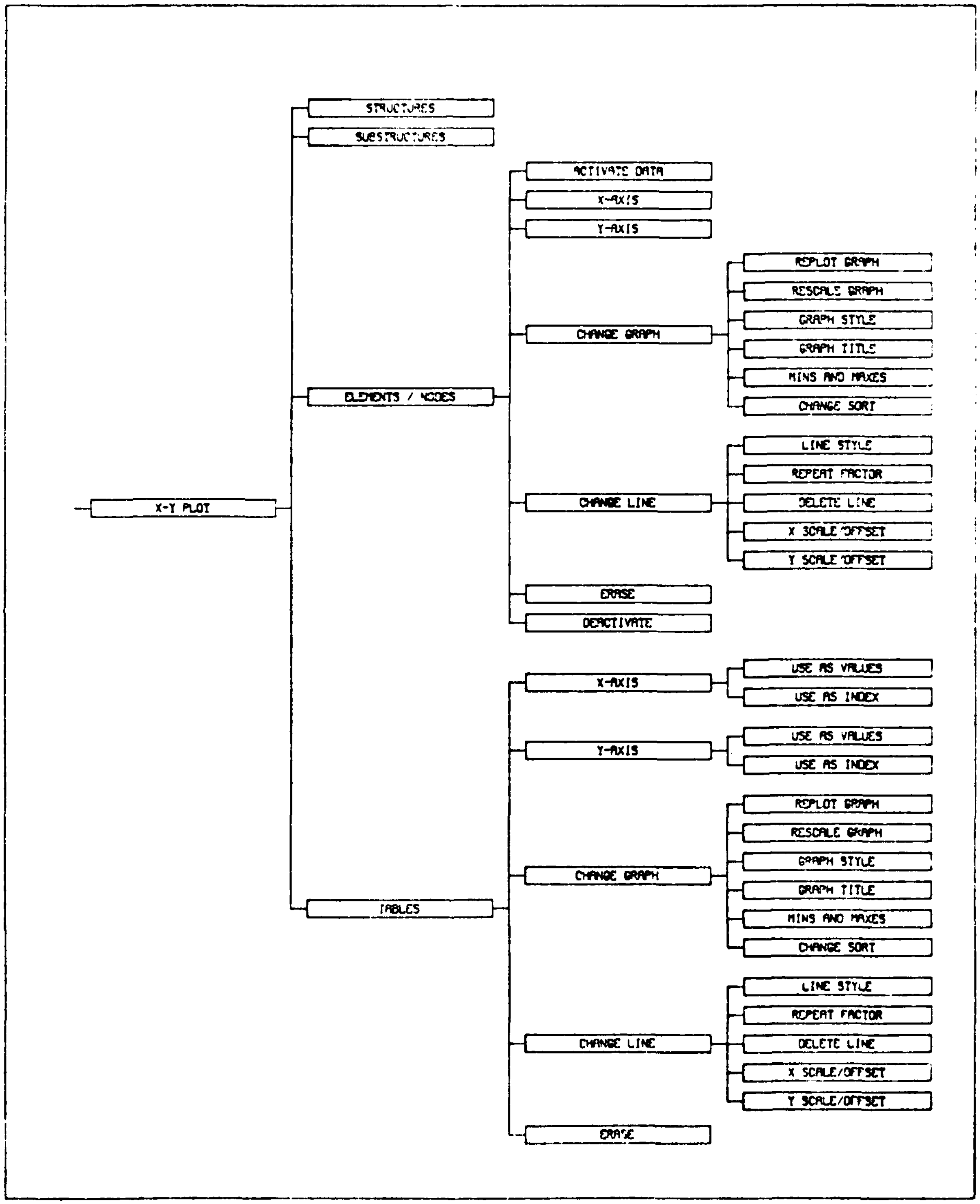


FIGURE 7.4.b. THE STAGING POSTPROCESSOR TREE

CHAPTER 8
GENERAL UTILITIES

This chapter of the STAGING User Manual describes several functions which have general utility during most STAGING sessions. These functions are located under the GLOBAL button which is always displayed. (In addition to the functions discussed in this chapter, there are other functions under GLOBAL which are discussed in Chapter 4.) The capabilities provided by the functions below include terminating the STAGING session, cataloging the current data base, displaying statistics about either the data base or the picture, obtaining information about the menu structure, and directly issuing INTERCOM commands.

8.1 STOP

The STOP function signals that the user is finished with the current STAGING session. The system closes all files and terminates the session. But before STOP actually stops processing, it requests that the user confirm that he is finished.

8.1.1 CONFIRM STOP COMMAND

CONFIRM STOP COMMAND is a checkpoint that allows the user to confirm the previously selected STOP request. If the user has not cataloged his data base (See Section 5.1), he can select RETURN and catalog his data base. When the user selects CONFIRM STOP COMMAND, the session is terminated.

8.2 CATALOG AS CURRENT

CATALOG AS CURRENT saves the data base from a STAGING session as a permanent file with the same name as that attached when the session began. The highest cycle of the current permanent file is automatically purged. If the user wishes to catalog an additional cycle, he uses CATALOG AS NEW. (Additional information on managing data base changes is contained in Section 5.1).

8.3 CATALOG AS NEW

CATALOG AS NEW allows the user to specify a new permanent file name or new cycle number for cataloging the current STAGING Data Base. When CATALOG AS NEW is selected a input template is displayed and the user is requested to change any or all of the data items. (Refer to Section 5.1 for more information on managing data base changes.)

8.4 DATA BASE STATISTICS

This function displays selected information about each level in the current data base. When DATA BASE STATISTICS is selected, the screen

is automatically erased and the information is printed in the DISPLAY AREA. This information includes:

1. Number of structures and their names
2. Number of substructures and their names
3. Number of elements, the range of element numbers, and a list of element attributes
4. Number of nodes, the range of node numbers, and a list of node attributes
5. Number of tables and their names.

8.5 DISPLAY STATISTICS

This function displays certain statistics about the current picture. When DISPLAY STATISTICS is selected, the screen is automatically erased and the statistics are printed in the DISPLAY AREA. The statistics which are printed include:

1. Whether the picture is 2D or 3D
2. The limits of the picture space in X,Y, and Z.
3. The number of active items at each level of the data base
4. The number of active items currently being displayed.

8.6 HELP

The HELP function displays information about the menu buttons and their relationships. When HELP is selected, the user can select from the four options discussed below.

8.6.1 DEFINE BUTTON

DEFINE BUTTON displays information about a user selected menu button or function. (This feature is not implemented as of 4/79.)

8.6.2 HISTORY OF PICKS

HISTORY OF PICKS prints a listing of the menu selections made by the user since entering a major module.

8.6.3 NEXT LEVEL UP

NEXT LEVEL UP displays the menu buttons on the next level up the menu tree. This is the menu the user will see if he selects RETURN on the current menu.

8.6.4 NEXT LEVEL DOWN

NEXT LEVEL DOWN displays the menu buttons on the next level down.

8.7 INTERCOM

The INTERCOM function is an interface to the INTERCOM system. It allows the user to execute a subset of the INTERCOM commands. Commands such as RETURN, RETAIN, and LOGOUT are not allowed.

This function can be used to batch conversion programs without terminating the current STAGING session. It is also useful for intergrating the status of existing jobs.

When the user selects INTERCOM, the screen is automatically erased and the following message is displayed:

* ENTER SCOPE CONTROL CARD.*

* TO EXIT, TYPE BYE OR END.*

* TYPE PAGE TO ERASE SCREEN.*

* --- *

The "*---*" is the prompt line and will appear after the successful execution of each command.

WARNING: If a control card error is detected by the INTERCOM function (such as a permanent file abort), it causes an immediate termination of the STAGING session. Therefore, INTERCOM should be used cautiously.

8.8 SUMMARY

The GLOBAL commands embodied in STAGING contribute greatly to its enormous flexibility and ease of use. When combined with the generation and display capabilities of other STAGING modules, these capabilities come very close to providing the user with analysis at his fingertips. Figure 8.1 shows the STAGING Utilities Tree.

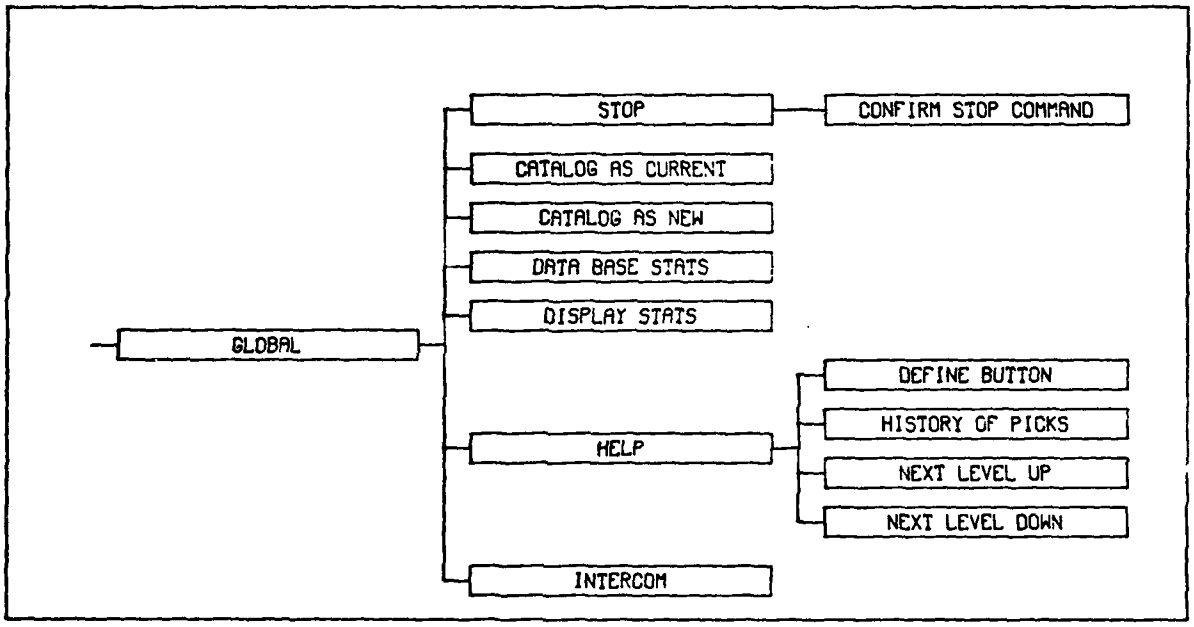


FIGURE 8.1. THE STAGING UTILITIES TREE

REFERENCES

(1) "The NASTRAN Users Manual" by McCormick, NASA SP 222 (01) (May, 1973)

(2) "The NASTRAN Theoretical Manual" by R. H. McNeal, NASA SP 222 (01) (May, 1973)

(3) "Finite Element Streee Analysis of Axisymmetric Solids" by E. L. Wilson and R. M. Jones, Air Force Report No. BSD-TR-67-228, (September, 1967)

(4) DOASIS - A Computer Code for the Deformation Plastic, Orthtropic Axisymmetric (and Plane) Solution of Inelastic Solids. Volume I: Finite Element Program, Theoretical and Programmers Manual. AFML TR 25-37, Vol I. (October, 1975)

(5) Volume II: Pre- and Post-Processor Computer Programs, Technical and Programmers Manuals, AFML TR 75-37, Vol II. (October, 1975)

(6) Volume III: Users Manuals, AFML TR 75-37, Vol III. (October, 1975)

(7) "HONDO, a Finite Element Computer Program for the Large Deformation Dynamic Response of Axisymmetric Solids" by S. W. Key, SANDIA Laboratories Report SLA-74-0039 (April, 1974)

(8) "An Automated Procedure for Flutter and Strength Analysis and Optimization of Aerospace Vehicles" Volume I. Theory and Applications, AFFDL TR-75-137, Vol I. Air Force Flight Dynamics Laboratory (December, 1975)

(9) "... See FAST01 " Volume II, Program Users Manual, AFFDL TR 75-137. Vol II. Air Force Flight Dynamics Laboratory (December, 1975)

(10) "ADINA - A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis" by Bathe, K. J. Massachusetts Inst. of Tech. Report No. 82448-1 (September, 1975)

*U.S.Government Printing Office: 1980 — 657-084/70

LMED

8